LAHDEN AMMATTIKORKEAKOULU
*Lahti University of Applied Sciences*

# VERKONVALVONTAJÄRJESTELMÄ

PK-yrityksen tarpeeseen

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikka
Tietoliikennetekniikka
Opinnäytetyö
Syksy 2011
Jere Uotila

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

UOTILA, JERE:

Verkonvalvontajärjestelmä
PK-yrityksen tarpeeseen

Tietoliikennetekniikan opinnäytetyö, 45 sivua, 13 liitesivua

Syksy 2011

TIIVISTELMÄ

---

Tämä opinnäytetyö käsittelee verkonvalvontajärjestelmän luomista aloittelevan PK-yrityksen tarpeeseen. Tekemällä itse sovelluksen yritys voi sijoittaa ohjelmistoihin suunnattuja pääomia muualle ja muokata sovelluksesta omaan tarpeeseen soveltuvan. Verkonvalvontajärjestelmän avulla yritys voi tarkkailla omia, tai asiakkaiden, verkossa sijaitsevien laitteiden kuntoisuutta, kuten linkkiyhteyksien tilaa tai prosessorin käyttöastetta.

Työn tuloksena saatiin toimiva verkonvalvontajärjestelmä. Verkonvalvontajärjestelmä toimii ilmaisen Linux-käyttöjärjestelmän päällä. Laitteiden tiedot haetaan automaattisesti Expect-ohjelmaa käyttäen joko telnet- tai SSH-protokollalla, tiedot tallennettaan MySQL-tietokantaan ja esitetään PHP:n avustuksella käyttäjälle. Verkkosivujen esittämisestä vastaa Apache-HTTP-palvelinohjelmisto. Käyttöliittymänä toimi verkkoselaimella käytettävä verkkosivu, joka toimii myös matkapuhelimella. Klikkaamalla sivulla olevaa linkkiä voidaan ottaa hallintayhteys laitteeseen joko telnet- tai SSH-pääteohjelmalla.

Saatujen tulosten perusteella voidaan todeta, että vapaaseen lähdekoodiin perustuvilla ohjelmistoilla voidaan luoda toimiva alusta verkonvalvontajärjestelmälle, ja itse tuotettu valvontasovellus toimii. Sovelluksen avulla voidaan nähdä yhdellä silmäyksellä laitteiden tilat, ja ongelman sattuessa voidaan nopeasti ryhtyä korjaamaan tilannetta.

Verkonvalvonnan avulla voidaan säästää sähköä. Tarkkailemalla laitteiden, kuten IP-puhelinten tai kytkimien, käyttöasteita voidaan luoda energian säästöön pyrkiviä käytäntöjä. Käytäntöjen avulla voidaan sammuttaa laitteet, kun työntekijät esimerkiksi poistuvat työtilasta. Sähköä säästämällä yrityksen sähkölasku pienenee ja autetaan säilyttämään maapallon uusiutumattomia energiavaroja.

Avainsanat: verkonvalvonta, FCAPS, Expect, PHP, MySQL, vapaan lähdekoodin ohjelmisto

Lahti University of Applied Sciences
Degree Programme Information Technology

UOTILA, JERE:

Network monitoring system
for small and medium-sized enterprises

Bachelor's Thesis in telecommunications technology, 45 pages, 13 appendices

Autumn 2011

ABSTRACT

---

This thesis deals with creating a network monitoring system for small and medium-sized enterprises which are new or entering the business. If a company creates network monitoring software by itself, the capital reserved for software purchases can be invested for other purposes. Another advantage is that the company can modify the software in a way that fits perfectly for the company's needs.

With the network monitoring system a company can monitor the status of the routers and switches in the company's own network or customers' devices at the edge of the network. The user interface shows the status of the network links and CPU usage.

Linux was chosen to be the operating system and other applications were also open-source software. The system logs in and obtains data from the monitored devices using an application called Expect, which writes a text file containing the whole conversation between the script and the monitored device. A PHP script parses the file and stores the needed data in to a MySQL database. A user sees the status of the devices on a web page, which retrieves the monitoring information from the database using another PHP script. By clicking a link on the web page the user can take control of the monitored device using a telnet or SSH client.

As for the results it can be said that making a network monitoring system using only open-source software is possible and relatively easy. In the testing environment the system worked as it should. At a glance it was easy to see the status of the monitored devices and in a problem situation the causes of the problem were easy to detect.

Key words: network monitoring, FCAPS, Expect, PHP, MySQL, open-source software

SISÄLLYS

LYHENNELUETTELO

| | |
|---|---|
| CLI | Command-line Interface, tekstipohjainen käyttöliittymä. |
| CSS | Cascading Style Sheets, WWW-dokumenteille kehitetty tyyliohjeiden laji. |
| DES | Data Encryption Standard, symmetrinen salausalgoritmi. |
| EOF | End-of-file, tiedoston loppumisen merkki. |
| FCAPS | Fault, Configuration, Accounting, Performance, Security, kehysrakenne tietoliikenneverkkojen ylläpitoon ja hallinnointiin. |
| GNU GPL | GNU General Public License, vapaiden ohjelmistojen julkaisemiseen tarkoitettu lisenssi. |
| HTML | Hypertext Markup Language, verkkosivujen teossa käytettävä kuvauskieli. |
| IP | Internet Protocol, TCP/IP-mallin protokolla. |
| IPTV | Internet Protocol Television, internetprotokollan käyttöön perustuva teknologia televisio-ohjelmien jakeluun. |
| LAMP | Linux, Apache HTTP Server, MySQL, PHP/Perl/Python, kokoelma avoimen lähdekoodin ohjelmia dynaamisia verkkosivuja tarjoavalle web-palvelimelle. |
| MD5 | Message-Digest Algorithm 5, kryptografiassa käytettävä viestitiivistefunktio. |

| | |
|---|---|
| MIB | Management Information Base, verkon hallinnassa käytettävä virtuaalinen tietokanta. |
| OID | Object Identifier, yksilöintitunnus, yleiskäyttöinen kansainvälisesti vain yhteen kohteeseen liitettävä numerosarja. |
| PHP | PHP: Hyperext Prepocessor, etenkin web-palvelinympäristöissä käytettävä ohjelmointikieli. |
| SHA | Secure Hash Algorithm, kryptografinen tiivistefunktio. |
| SNMP | Simple Network Management Protocol, TCP/IP-verkkojen hallinnassa käytettävä tietoliikenneprotokolla. |
| SSH | Secure Shell, salattuun tietoliikenteeseen tarkoitettu protokolla. |

# 1 JOHDANTO

Aloittelevalle yritykselle ohjelmistohankinnat voivat tulla kalliiksi. Tekemällä itse ja käyttämällä vapaita ohjelmistoja voidaan ohjelmistohankintoihin varattuja pääomia hyödyntää yrityksen kannalta muihin toimintoihin. Samalla ohjelmistosta voidaan muokata omaan käyttötarkoitukseen hyvin soveltuva ratkaisu.

Työssä luodaan verkonvalvontajärjestelmä, jolla valvotaan verkossa olevia laitteita. Valvontajärjestelmää voidaan käyttää esimerkiksi yrityksen sisäisesti omien verkkolaitteiden valvomiseen, tai jos yritys toimii palveluntarjoajana, valvontajärjestelmällä voidaan valvoa verkon rajalla olevia asiakkaiden laitteita. Halutessaan valvontajärjestelmään voidaan luoda oma käyttöliittymä asiakkaiden käyttöön, jonka avulla asiakas voi tarkastella omien laitteidensa tilaa.

Tutkimusongelmana tässä työssä on toteuttaa verkonvalvontajärjestelmä pienen tai keskisuuren yrityksen tarpeisiin. Ensimmäisenä tavoitteena on löytää tietoturvallinen ja vapaisiin ohjelmistoihin perustuva alusta järjestelmälle. Vapaat ohjelmat ovat maksuttomia, joten valvontajärjestelmän pystyttäminen ei tule maksamaan käyttäjälle ohjelmistojen osalta yhtään mitään. Toisena tavoitteena on luoda alustan päälle valvonnasta vastaava ohjelmisto, joka on käyttöjärjestelmä- ja laiteriippumaton, niin palvelin- kuin käyttäjäpuolella, helpohkosti muokattavissa omien tarpeiden mukaan ja helppo käyttää.

Työn teoriaosassa käsitellään ensin verkonvalvontaa käsitteenä tiedon hakemisesta laitteelta aina tiedon käyttäjälle esittämiseen asti ja tutkitaan, mitä tietoa voidaan ja kannattaa kerätä. Samalla vertaillaan kolmea erilaista verkonvalvonnan yhteyskäytäntöä, komentoliittymää, SNMP:tä (Simple Network Management Protocol) ja Syslogia, joiden avulla laitteiden valvontatietoja voidaan kerätä ja esittää käyttäjälle. Työn käytännön osassa luodaan valvontajärjestelmä käyttöjärjestelmän asennuksesta lähtien.

# 2 VERKONVALVONTA

## 2.1 Yleistä verkonvalvonnasta

Yritykset ovat entistä riippuvaisempia tietoliikenteestä, jolloin pienikin verkon häiriö saattaa aiheuttaa mittavia tappioita yrityksen toiminnalle. Tästä syystä verkonvalvonta on oleellinen osa ylläpidettävän verkon hallintaa. Verkon tilan jatkuvalla tarkkailulla saadaan selville mahdolliset ongelmat tietoliikenneverkon toiminnassa, ja näin pystytään tarttumaan ongelmiin ennen kuin ongelmat saattavat lamauttaa koko yrityksen toiminnan. Valvonnalla pystytään ongelmien havaitsemisen lisäksi ennustamaan mahdollisia päivitystarpeita tutkimalla esimerkiksi verkkoliityntöjen käyttöastetta. (Clemm 2007, 6 - 8.)

Verkonvalvonta voidaan ymmärtää käsitteenä, jossa haetaan tietoja järjestelmän tilasta ja laitteiden määrittelyistä, ja yhdistetään saadut tiedot yhteen kokonaisuuteen. Saatuja tietoja voidaan käyttää järjestelmän tilan valvomiseen, jalostaa jatkokäyttöä varten tai tiedoista voidaan luoda raportteja. (Verma 2009, 112.)

Verkonvalvonta osa verkonhallintaa. Verkonhallinta koostuu eri osista, jotka määrittelevät verkon toiminnan, hallinnan, ylläpidon ja verkon tarjoamisen käyttäjille. Tavoitteisiin päästään ylläpitämällä verkkoa ja verkon palveluita siten, että ylläpitäjällä on reaaliaikainen kuva verkon toiminnasta, ja vikatilanteen sattuessa asiakkaille koituva häiriö jää mahdollisimman pieneksi. Häiriöitä vastaan taistellaan hallinnalla ja ylläpidolla, jolloin ylläpitäjä tietää tarkasti, mistä verkko koostuu, päivittää verkon laitteita tehokkaammiksi ja mukautuu verkon muutoksiin konfiguroimalla laitteita uudelleen. Muita hallintaan kuuluvia osa-alueita ovat muun muassa verkon suunnittelu, organisointi, kirjanpito ja tietoturvasta huolehtiminen. (Ding 2009, 43 - 44.)

## 2.2 Verkonvalvonnan geneerinen malli

Vaikka markkinoilla on useita verkonvalvontaan tarkoitettuja ohjelmistoja ja järjestelmiä, lähes kaikki toimivat saman mallin mukaan. Mallia voidaan kuvata viisiportaisena geneerisenä mallina. (Verma 2009, 112.)

Mallissa tason alimmaisena on tiedon kerääminen valvottavasta laitteesta, tiedon jalostaminen tietokantaa tallentamista varten, itse tietokanta ja tiedon hakeminen tietokannasta jatkokäyttöä varten. Tiedon raportointi järjestelmän käyttäjälle on mallissa viimeisenä. (Verma 2009, 113.)

### 2.2.1 Tiedon kerääminen

Tiedon kerääminen on ensimmäinen askel monitorointiprosessissa. Laitteista pyritään hankkimaan erilaista haluttua informaatiota, jota voidaan tallentaa tietokantaan. Tiedon keräämisessä on kaksi ongelmaa, jotka on otettava huomioon valvontasovelluksen teossa: skaalautuvuus ja valvottavien laitteiden heterogeenisyys. Yritysten verkoissa saattaa olla tuhansia laitteita, joita pitää valvoa. Valvontatapahtuma ei kuitenkaan saisi häiritä millään tavoin verkon toimintaa tai valvottavia laitteita, jolloin on suunniteltava, kuinka usein ja kuinka paljon valvottavaa tietoa halutaan kerätä. Toisena ongelmana on verkossa olevien laitteiden heterogeenisyys. Jokaista laitetyyppiä varten on luotava oma toimintatapa, jolla tietoa kerätään, koska eri laitteet saattavat tuottaa valvottavaa tietoa eri tavalla ja eri muodossa. (Verma 2009, 114.)

Tietoa voidaan kerätä joko passiivisesti tai aktiivisesti. Passiivisessa tavassa valvontaohjelmisto ei itse aktiivisesti pyydä mitään tietoja valvottavalta laitteelta, vaan ottaa ainoastaan vastaan valvottavan laitteen lähettämät viestit. Esimerkkinä passiivisesta keräämisestä on SNMP trap -viestit. Aktiivisessa keräämisessä valvontaohjelmisto pyytää valvottavalta laitteelta tietoja, joihin laite vastaa. Esimerkkinä aktiivisesta keräämisestä ovat ping- ja traceroute-komennot, joilla voidaan selvittää valvottavan laitteen saavutettavuus. Aktiivisen keräämisen hyviä puolia ovat muun muassa jumiutuneen laitteen tunnistaminen ajoissa ja tietojen

hakeminen halutessa, mutta huonona puolena on ylimääräisen liikenteen tuottaminen. (Claise & Wolter 2007, 103, 122.)

### 2.2.2 Tiedon jalostaminen tietokantaan tallentamista varten

Tiedon keräämisen jälkeen tietoa on jalostettava ennen tietokantaan tallentamista, koska hankittu tieto saattaa olla virheellistä, puutteellista tai tietoa on liian paljon. Tietokantaan tallennettaessa on varmistuttava, että tallennettu tieto on oikeata ja oikeassa muodossa. (Kamber 2006, 48)

Suurta verkkoa ylläpidettäessä valvontatietokantojen koko saattaa kasvaa suuriin mittoihin hyvin nopeastikin. Suurien tietokantojen ylläpito ja hallinnointi voi olla kallista, joten turhan tiedon poistaminen ja pakkaaminen voi olla hyödyllistä. Jo ennen tietokantaan tallentamista kerättyä tietoa voidaan suodattaa reaaliajassa. Valvonta-ohjelmalla saattaa olla palvelimen keskusmuistissa pienempi versio käytettävästä tietokannasta, johon ohjelma vertaa hankittua tietoa poistaen ylimääräisen tiedon. Menetelmiä reaaliaikaiseen tiedon vähentämiseen on pääasiassa kolme. Yhdistämällä ja keskiarvoistamalla useita saatuja tietoja, kuten reitittimen prosessorin kulutus pidemmällä aikavälillä tai yhtä web-sivua pyörittävien useiden palvelinten kaistankulutus, voidaan vähentää tietokantaa kohdistuvaa rasitusta. Tietoja voidaan tallentaa tietokantaan vasta kun jokin tietty raja-arvo ylittyy, esimerkiksi prosessorin käyttöaste nousee yli 60 prosentin. Jos raja-arvo ei ylity, mitään tai joitain tietoja ei tallenneta, raja-arvon ylittyessä kaikki tarvittavat tiedot kerätään talteen. Kaksoiskappaleiden ja ylimääräisten tiedonkappaleiden poistaminen vähentää myös tietokannan rasitusta. Esimerkiksi reitittimen vikaantuessa reitittimeen liitetyt naapurilaitteet voivat alkaa lähettää virhesanomia vikatilanteesta. Useita samanlaisia viestejä ei ole järkevää varastoida. (Verma 2009, 123 - 124.)

Talteen otettu tieto saattaa olla puutteellista, korruptoitunutta tai epäjohdonmukaista. Tiedon puhdistamisella pyritään varmistamaan, että tietokantaan tallennettava tieto on validia. Valvontajärjestelmään saattaa tulla vika, jonka seurauksena halutut muuttujat eivät tallennu tietokantaan. Tietokantaan ei aina voi jättää puuttuvia arvoja, jolloin tallennettavan arvon puuttuessa arvo voidaan korvata jollakin

muulla tiedolla. Arvo voidaan syöttää käsin, mutta työ vaatii aikaa, eikä ole aina mahdollista puuttuvien arvojen suuren määrän vuoksi, arvo voidaan korvata tunnetulla vakiolla, kuten sanalla "tuntematon" tai muuttujan arvoksi voidaan syöttää todennäköisin arvo. (Kamber 2006, 61 - 62.)

Prosessina tiedon puhdistaminen alkaa epäjohdonmukaisuuksien havaitsemisella. Epäjohdonmukaisuudet havaitaan parhaiten tietämällä, miltä kerätyn tiedon pitäisi näyttää, esimerkiksi mitkä ovat mahdolliset arvot, joita muuttuja voi saada, tai ovatko arvot tilastollisesti mahdollisia. Jos arvot poikkeavat halutuista, on oltava suunnitelma, miten saatuja arvoja käsitellään. Yritetäänkö silloin hakea arvot uudelleen, tiedon tallentaminen keskeytetään vai syötetäänkö tietokantaan vakiomuuttujat? (Kamber 2006, 65 - 67.)

### 2.2.3 Tietokanta

Tietokannat ovat pysyviä tietovarastoja, joihin tallennetaan tietoa. Tietokantoja voidaan käyttää esimerkiksi välimuistina, johon tallennetaan valvottavien laitteiden tietoja. Tällöin tieto on helposti saatavilla eikä verkkoa tai valvottavaa laitetta tarvitse kuormittaa ottamalla yhteyden ja hakemalla tiedot. Laskutus voi pitää tietokannassa tiedot asiakkaista ja käytetyistä palveluista, jolloin laskujen tuottaminen on helppoa. (Clemm 2007, 31.)

Tietokannan rasitus kasvaa ajan myötä, joten jo suunnitteluvaiheessa on kiinnitettävä huomiota skaalautuvuuteen. Tietokannalla täytyy olla riittävästi tallennustilaa tietoja varten, ja järjestelmän on pystyttävä käsittelemään tietokantaan kohdistuvat luku- ja kirjoituspyynnöt. (Verma 2009, 129 - 130.)

### 2.2.4 Tiedon hakeminen tietokannasta ja raportointi

Valvontasovelluksesta riippuen tietokantaan kerättyä tietoa voidaan käyttää kahdella tapaa hyväkseen. Suoraan tulostettuna tai erillisen diagnostiikkamoduulin kautta. (Verma 2009, 138.)

Kun tietokanta tulostetaan kokonaisena esimerkiksi tietokoneen näytölle, on helppo nähdä, missä viat sijaitseva. Tällöin ylläpitäjä voi ryhtyä itse korjaustoimiin. Lisäämällä valvontajärjestelmään erillisen diagnostiikkamoduulin valvontatapahtumia voidaan automatisoida ja korjausaikoja lyhentää. Moduuli tutkii tietokantaan tallennettuja muuttujien arvoja ja vian sattuessa pyrkii selvittämään syyn vikaan. Kun syy on selvillä, järjestelmä voi eristää viallisen osan verkosta ja ryhtyä korjaustoimiin. Jos automaattikorjaus ei onnistu, verkon ylläpitäjää informoidaan tapahtuneesta. (Verma 2009, 138 - 139.)

# 3 VERKONVALVONNAN YHTEYSKÄYTÄNNÖT

## 3.1 Yleistä yhteyskäytännöistä

Verkossa olevien laitteiden valvontaa voidaan suorittaa usealla eri tavalla, yhteyskäytännöllä. Yhteyskäytäntöjen avulla laitteet kommunikoivat tilankerääjien kanssa pyytäen ja lähettäen tietoja. (Clemm 2007, 249.)

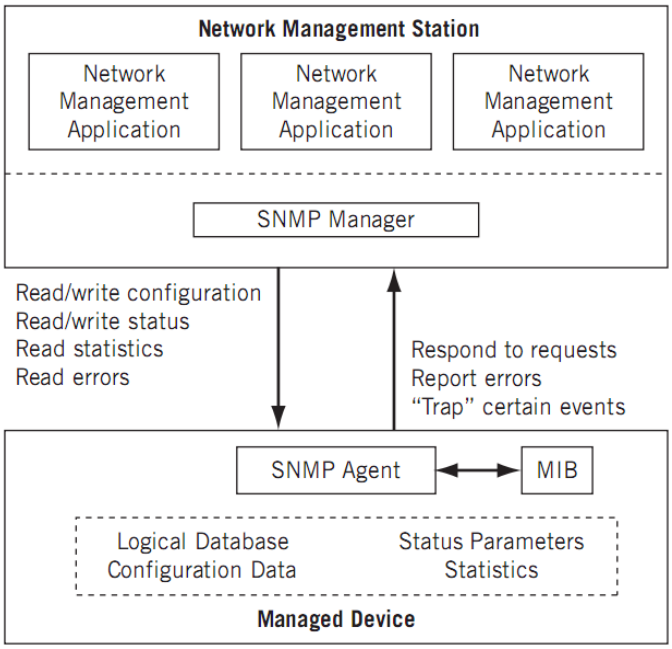
Taulukosta 1 selviää, että osa protokollista sopii paremmin ihmiskäyttäjän ja valvottavan laitteen väliseen kommunikointiin. Toinen osa protokollista on tarkoitettu valvottavan laitteen ja laitetta valvovan ohjelman väliseen tiedonsiirtoon. (Clemm 2007, 249.)

TAULUKKO 1. Hallintaprotokollien asemointi ihmiskäyttäjien ja sovellusten kesken (Clemm 2007, 209)

| User / Application | Humans | Applications |
|---|---|---|
| Monitoring | CLI, syslog | SNMP, syslog |
| Configuration | CLI | Netconf |
| Data Collection | n.a. | Netflow/IPFIX |

## 3.2 SNMP

SNMP on kenties tunnetuin verkonvalvonta- ja verkonhallintaprotokolla, joka perustuu asiakas / palvelin -arkkitehtuuriin. SNMP:ssä asiakasta kutsutaan hallinta-agentiksi, lyhyesti pelkäksi agentiksi, ja palvelinta hallinta-asemaksi. Hallinta-asema suorittaa loppukäyttäjälle näkyvää verkonvalvontasovellutusta, ja agentti on jokaisessa valvottavassa laitteessa omassa käyttöjärjestelmässään itsenäisesti suoritettava ohjelma. Hallinta-asema lähettää käskyjä agenteille, jotka suorittavat käskyn, kuittaavat käskyn suoritetuksi hallinta-asemalle ja jäävät odottamaan lisäkäskyjä. Valvonta-asema tallentaa tietokantaansa käskyn lopputuloksen. Valvottavassa verkossa on siis yleensä yhdestä muutamaan hallinta-asemaa ja useita agentteja, eli valvottavia kohteita. Kuviosta 1 selviää SNMP:n toimintamalli. (Goralski 2009, 616.)




KUVIO 1. SNMP:n toimintamalli (Goralski 2009, 617)

SNMP-laitteet keskustelevat keskenään viidellä erilaisella operaatiolla: get, get-next, set, get-response ja trap. Get-pyynnöllä hallinta-asema pyytää tietoja, MIB-objekteja (Management Information Base), agentilta. Tällainen tieto voi olla vaikkapa ulkoverkon liitymän IP-osoite. Get-next-pyyntö toimii samalla tapaa kuin

get-pyyntökin sillä erotuksella, että agentti palauttaa pyydetystä objektista poiketen puumaisessa MIB-tietokannassa seuraavana, leksakrografisessa järjestyksessä, olevan objektin. Objektit erotetaan omalla uniikilla osoitteellaan, jota kutsutaan OIDiksi (object identifier). Valvottavan laitteen isäntänimi saattaa olla vaikka MIB:ssä OID-tunnuksella 1.3.6.1.4.1.2011.1. Komentoa voidaan käyttää esimerkiksi siinä tilanteessa, kun käyttäjä ei tiedä mitä objekteja hallittavan laitteen MIB-tietokannassa on. Get-next-komennolla pääsee "kävelemään" MIB-tietokannassa järjestyksessä alusta loppuun selvittäen agentin MIB-tietokannan rakenteen. Set-pyynnöllä hallinta-asema pyytää muuttamaan jonkin objektin hallittavan laitteen MIB-tietokannassa. Set-pyynnöllä voidaan vaikkapa vaihtaa hallittavan laitteen isäntänimi. Agentti lähettää get-response-viestin vastauksena hallinta-aseman pyyntöihin. (Clemm 2007, 251 - 256.)

Vakavan ongelman ilmetessä, kuten verkkoyhteyden katketessa, valvottava laite voi itsenäisesti ilmoittaa hallinta-asemalle ongelmasta niin sanotun trap-viestin avulla. Tällöin verkon ylläpidosta vastaava taho voi ryhtyä välittömästi korjaamaan vikaa. (Goralski 2009, 617.)

Valvottavassa laitteessa on MIB, eli hierarkkinen tietokanta, jota hallinta-agentti käyttää laitteen tietojen hakemiseen tai tietojen muuttamiseen. MIB ei kuitenkaan ole tietokanta sanan varsinaisessa tarkoituksessa, koska MIB ei itsessään sisällä mitään tietoja valvottavasta laitteesta. MIB on eräänlainen kerros SNMP:tä käyttävän valvontasovelluksen ja valvottavan laitteen rekistereiden, joissa on esimerkiksi tieto ulkoverkon liitynnän tilasta, välillä. Esimerkiksi verkonvalvonnasta vastaava taho haluaa tietää valvottavan laitteen prosessorikulutuksen. Taho lähettää get-pynnön valvottavan laitteen hallinta-agentille, joka vertaa pyynnössä ollutta muuttujaa MIB-tietokannassa olevaan vastaavaan laitteen rekisterin muuttujaan. Rekisterissä ollut muuttujan arvo lähetetään get-response-vastauksella hallinta-asemalle. (Clemm 2007, 173.)

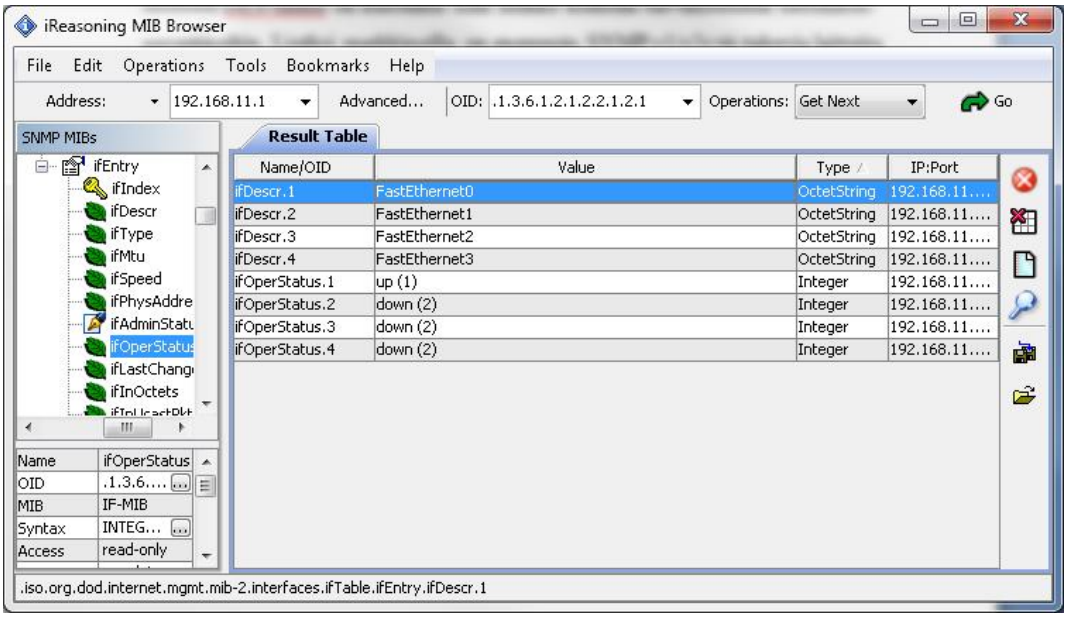
Koska MIBit sijaitsevat valvottavissa laitteissa, valvonta-asemat eivät voi tietää kaikista laitteista mitä mikäkin MIBin osoite tarkoittaa valvottavaan laitteen rekisterissä. Tästä syystä valvottavien laitteiden valmistajat ja valvontasovelluksien

tekijät julkaisevat MIB-määrittelyjä, jotka parittavat valvontasovelluksen MIBin ja valvottavan laitteen MIBin vastaaviksi. (Clemm 2007, 181.)

SNMP voi olla järjestelmään hyökkäävälle kultakaivos. Joissain laitteissa SNMP saattaa olla lukuoikeuksilla oletuksena päällä mahdollistaen hyökkääjälle keinon kartoittaa koko valvottavan verkon rakenne. Tunkeutuja voi saada selville esimerkiksi verkossa olevien laitteiden mallit, IP-osoitteet, ohjelmistojen versiotiedot. Saatujen tietojen avulla on mahdollista etsiä järjestelmän haavoittuvuuksia. (Akin 2002, 68.)

Nykyään käytössä on kolme SNMP:n versiota. 1990-luvun alussa SNMP v1:stä tuli standarditapa valvoa verkossa olevia laitteita. SNMP v1 on kuitenkin tietoturvaltaan kehnoa tasoa. SNMP-viestien autentikoitiin käytetään ryhmätunnusta, joka kulkee tietoliikenneverkoissa salaamattomana eli selväkielisenä. Tunnuksen selvittäminen on helppoa esimerkiksi protokolla-analysaattorilla, joten tunnuksen selvittämisen jälkeen tunkeutujalla on pääsy SNMP v1:stä käyttäviin laitteisiin. Tietoturvaa yritettiin parantaa SNMP v2c:n avulla, mutta standardista päättävät tahot eivät päässeet yhteisymmärrykseen tietoturvan saralla. SNMP v2c sisältää v1:een nähden parannuksia ainoastaan muilla osa-alueilla kuin tietoturvan, eli vihamielisellä tunkeutujalla on mahdollisuus selvittää selväkielisenä kulkeva ryhmätunnus ja tunkeutua järjestelmiin selvitetyn tunnuksen avulla, kuten SNMP v1:n kohdalla. (Akin 2002, 69.)

SNMP v3 toi lopulta parannuksia tietoturvaan. Autentikointiin voidaan käyttää MD5- (Message-Digest Algorithm 5) tai SHA-tiivisteitä (Secure Hash Algorithm), datapaketti voidaan salata kokonaan DES-lohkosalauksella (Data Encryption Standard) ja viestin eheys voidaan taata. Useiden tietoturva-ammattilaisten mielestä DES-salaus on kuitenkin liian heikko korkean turvallisuuden tietoliikenneverkkoihin. Lisäksi markkinoilla on enemmän SNMP v1/v2c:tä tukevia laitteita kuin v3:sta. (Akin 2002, 70.)



KUVIO 2. SNMP-ohjelman käyttöliittymä

Kuviossa 2 on iReasoning MIB Browser -ohjelman käyttöliittymä. Ohjelmaa käyttämällä on haettu SNMP get next -toiminnolla reitittimeltä neljän ensimmäisen verkkoliittymän kuvaus ja tämän jälkeen kyseisten liittymien toiminnallinen tila.

## 3.3 CLI eli komentoliittymä

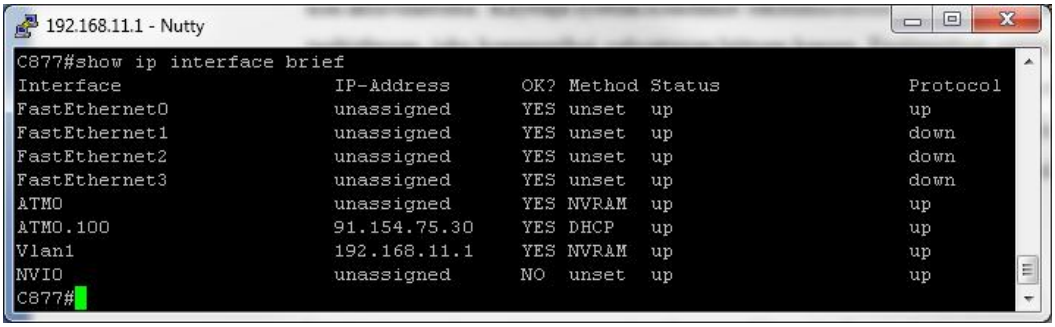
Komentoliittymä kehitettiin, jotta käyttäjät pääsisivät helposti hallinnoimaan verkon aktiivilaitteita. Käyttäjä syöttää komennot tekstimuodossa tietokoneella pääteohjelmaan, joka kommunikoi valvottavan laitteen kanssa. Ensimmäiset reitittimet olivat palvelimia, joissa käyttöjärjestelmänä oli Unix. Nykyään reitittimet eivät suurpiirteisesti eroa ensimmäisistä malleista. Reititin on erityiseen tarkoitukseen rakennettu tietokone usealla verkkoliitynnällä ja tarkoitukseen ohjelmoidulla käyttöjärjestelmällä. (Clemm 2007, 261.)

Komentoliittymän käyttämiseen vaaditaan omat ohjelmansa, jotka pääosin ovat ilmaisia. Laitteesta riippuen laitetta voidaan hallinnoida paikallisesti, etänä tai sekä että. Paikallisesti hallittavan laitteen ja tietokoneen välille kytketään kaapeli, ja käynnistetään hallinnointiin tarkoitettu ohjelma. Ciscon laitteissa tietokoneen sarjaportin ja laitteen välille kytketään konsolikaapeli ja itse laitetta hallinnoidaan

pääte-emulaattorilla, kuten Tera Termillä. Etänä hallittaviin laitteisiin käytetään pääosin SSH- (Secure Shell) tai telnet-protokollaa tukevia asiakasohjelmia, kuten PuTTYa. (Neumann 2009, 2 - 3.)

Yhtä yhteistä standardoitua komentoliittymä ei ole olemassa, vaan valmistajat käyttävät tuotteissaan itse tuottamiaan komentoliittymiä. Jopa valmistajan omissa tuotteissa saattaa olla toisistaan poikkeavia liittymiä. Tästä syystä yhden valmistajan komentoliittymässä toimivat komennot eivät välttämättä toimi toisen valmistajan järjestelmässä. Kuitenkin toimintaperiaatteet ovat eri järjestelmissä samoja, jolloin yhden järjestelmän tunteminen auttaa muiden valmistajien järjestelmien käyttämisessä. (Clemm 2007, 261 - 262.)

SNMP:n tai vastaavien yhteyskäytäntöjen kautta ei aina voida suorittaa kaikkia valvottavaan laitteeseen kohdistuvia valvontatoimenpiteitä, jolloin on turvauduttava varsinaiseen hallinta- / valvontaliittymään eli komentoliittymään. Komentoliittymä kehitettiin alunperin ihmiskäyttäjiä varten kommunikoimaan suoraan hallittavan laitteen kanssa, joten komentoliittymän käyttö osana erillistä käyttöliittymää tuo omat haasteensa. Komentojen syöttäminen valvottavalle laitteelle ei ole ongelma, vaan palautuvien tulosten jäsentäminen. Tulokset on yleensä jäsennelty siten, että ihmiskäyttäjälle tulosten tulkitseminen on helppoa. Konekäyttäjälle saaduista tuloksista on usein eroteltava oleelliset tiedot muun tekstin seasta erilaisia skriptejä käyttäen. ( Clemm 2009, 265 - 266.)




KUVIO 3. Komentokäyttöliittymä

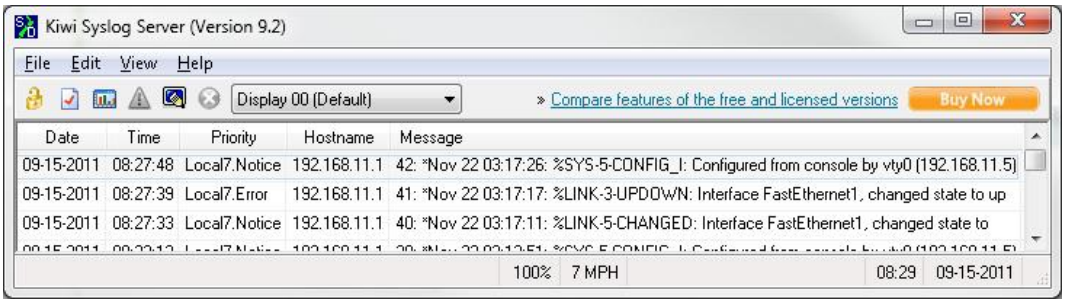
Kuviossa 3 Cisco-reittimen komentokäyttöliittymässä annetaan komento *show ip interface brief*. Komennon tuloksena saadaan selville perustietoja verkkoliityntöjen tilasta ja IP-osoitteet (Internet Protocol). Esimerkiksi liitynnän ATM0.100 IP-osoite on 91.154.75.30, ja liityntä on toiminnassa. (Neumann 2009, 8.)

Show-komennon tuloksena saadun taulukon tulkitseminen on ihmiselle helppoa. Tietyn kohdan, kuten verkkoliitynnän IP-osoitteen, havaitsee helposti. Sovellukselle ymmärtäminen ei ole yhtä helppoa, vaan sovelluksen on tiedettävä missä muodossa ja kohtaa tulosta haluttu tieto sijaitsee. Tuloksen jäsentelyä varten on luotava oma algoritmi. Sovellus näkee tuloksen merkkijonoina, välilyönteinä, sarkaimina ja rivinvaihtoina, joiden avulla halutun tiedon hankkiva algoritmi on luotava. Onneksi laitevalmistajat pyrkivät pitämään tietyn komennon, kuten Cisco IOS:ssä käytettävän *show ip interface brief*:n, tuloksen rakenteen samankaltaisena laitteesta riippumatta. Tällöin algoritmin käyttäminen halutun tiedon hankkimiseksi saman valmistajan eri laitteissa vaatii vähän muutoksia itse algoritmin ohjelmakoodiin. (Clemm 2009, 266 - 267.)

## 3.4 Syslog

Syslog on standardi järjestelmäviestien kirjaamisessa lokitiedostoon. Valvottava kohde lähettää itsenäisesti tapahtumista, kuten kriittisistä yhteyksien katoamisista aina tavalliseen debuggaustietoon viestejä lokiin tallennetavaksi. Lokitiedosto voi sijaita valvottavassa laitteessa itsessään, jolloin verkonvalvonnasta vastaava taho hakee lokitiedoston valvottavasta laitteesta jatkotoimenpiteitä varten, tai valvottava laite voi lähettää syslog-viestit suoraan valvonta-asemalle. Jotkut laitteet tuottavat niin suuria määriä viestejä, ettei suurimmasta osasta viestejä ole käytännön hyötyä. Kuitenkin keräämällä lokitietoja voidaan esimerkiksi selvittää mahdolliset murtautumisyritykset laitteisiin, tai etsiä syitä selittämättömiin verkkoyhteyksien katkoksiin. Syslog-viesteistä ei lähetetä kuittausta vastaanottajalle, joten viestien eheyden tarkistaminen ei ole mahdollista. Lisäksi yhteyden katkeaminen valvottavan laitteen ja valvonta-aseman välillä aiheuttaa viestien häviämisen, ellei niitä tallenneta paikallisesti valvottavaan laitteeseen. (Clemm 2009, 268.)

Syslog, kuten komentoliittymäkin, kehitettiin ihmiskäyttäjiä varten. Viestit ovat ihmisille helppoja ymmärtää, mutta ohjelmia varten viestin sisältöä on jäsenneltävä halutun tiedon esiin saamiseksi. Syslog-viesteissä on kaksi osaa: viestin otsikko ja viestin runko. Otsikko-osan tiedot kertovat järjestelmällisesti tietoja viestistä, kuten viestin lähettämisajan ja lähettäjän, tapahtuman vakavuuden, viestin lähettäneen alijärjestelmän ja viestin tyypin. Alijärjestelmällä tarkoitetaan esimerkiksi valvottavassa laitteessa ajettavaa prosessia tai ohjelmaa, joka on lähettänyt viestin. Alijärjestelmäluokkia on 24 kappaletta, joista 8 (local0 - local7) on varattu valmistajien vapaaseen käyttöön. Runko-osassa on selväkielinen viesti, kuten tieto verkkoliitynnän aktivoitumisesta. (Deveriya 2006, 142 - 145.)




KUVIO 4. Vastaanotettuja syslog-viestejä

Kuviossa 4 on reitittimeltä vastaanotettuja syslog-viestejä. Esimerkiksi toisella rivillä Date-sarakkeessa on päivämäärä, milloin hallinta-asema on vastaanottanut viestin ja Time-sarakkeessa on hallinta-aseman kellonaika viestin vastaanottoaikana. Priority-sarakkeesta selviää, että viestin lähettäjä on jokin vapaaseen alijärjestelmäluokkan kuuluva prosessi ja viestin vakavuus on arvoltaan virhe. Hostname-sarakkeesta selviää valvottavan laitteen IP-osoite, ja Message-sarakkeessa on itse viesti. Viestin alussa on viestejä erotteleva juokseva numero, paikallinen päivämäärä ja aika, jolloin valvottava laite on lähettänyt viestin. Prosenttimerkin jälkeen tule alijärjestelmän nimi, joka on tässä tapauksessa Cisco-spesifinen LINK, viestin vakavuus ja viestiä kuvaava tunniste. Tunnisteen jälkeen tulee selventävä teksti viestin sisällöstä. (Deveriya 2006, 142 - 145.)

# 4 FCAPS

## 4.1 Vikojen hallinta

FCAPS (Fault, Configuration, Accounting, Performance, Security) on ISO Telecommunications Management Network -protokollamalli ja kehys tietoliikenneverkkojen hallintaan. Suomennettuna FCAPS tarkoittaa vikojen, kokoonpanon, käytön, suorituskyvyn ja turvallisuuden hallintaa. (Ding 2009, 90.)

Vianhallintakäsitteeseen kuuluu muun muassa tietoliikenneverkossa olevien vikojen havainnointi, eristäminen ja korjaaminen, lokitietojen vastaanottaminen, tietojen tutkiminen ja tallentaminen sekä diagnostiikkatestien suorittaminen valvottaville laitteille. Verkosta vastaava taho valvoo verkkoa vianhallintakonsolin kautta, josta näkee hallittavan verkon tilan. Vian sattuessa vianhallintakonsoli korjaa ideaalitapauksessa vian itsenäisesti esimerkiksi ajamalla korjausskriptejä, tai hälyttää viasta ylläpitäjälle sähköpostin tai tekstiviestin avulla. (Ding 2009, 90.)

Vianhallinta voidaan suorittaa kahdella eri tavalla: passiivisesti tai aktiivisesti. Passiivisessa tavassa valvottava laite lähettää itse tietoja valvontaa suorittavalle asemalle esimerkiksi SNMP-viestien avulla. Jos kuitenkin valvottava laite jumiutuu täysin, jolloin valvottava laite ei voi edes lähettää viestejä tapahtuneesta virheestä, vika saattaa jäädä vianhallinta-asemalta huomaamatta. Suoritettaessa vianhallinta aktiivisesti valvonta-asema lähettää vaikkapa PING-pyyntöjä, joihin laite vastaa. Jos valvottava laite ei vastaa PING-pyyntöihin, valvonta-asema tekee hälytyksen. (Ding 2009, 90.)

Hälytysten keräämisen jälkeen ylläpitäjän saatavilla pitää olla ajantasainen ja tarkka lista hälytyksistä. Listan avulla saadaan selville, missä kunnossa valvottava verkko on ja mitkä palvelut eivät toimi niin kuin pitäisi. Listaa on muokattava sitä mukaa, kun vikoja on korjattu tai viat on todettu aiheettomiksi. Hälytyslistan on oltava ylläpitäjälle helppolukuinen, jotta vikojen sijainnit ja vakavuudet saadaan selville helposti. Yleinen tapa esittää verkon rakenne on topologiakartta. Kartassa laitteet on esitetty ikoneina ja laitteiden väliset yhteydet viivoina. Esille tulleet viat voidaan esittää animoinnilla tai värikoodauksella. Punaisella värillä

voidaan merkitä kriittiset viat, oranssilla merkittävät hälytykset, keltaisella vähäpätöiset hälytykset ja vihreä tarkoittaa, että vikoja ei ole. Harmaalla voidaan kuvittaa hallintayhteyden puuttumista. (Clemm 2009, 133 - 134.)

Ajantasaisen hälytystietokannan lisäksi hälytyksiä voidaan kerätä historiatietokantaan. Historiatietokannan avulla voidaan pitää kirjaa laitteiden luotettavuudesta ja kerätä tilastoja verkon toiminnasta. Lisäksi, jos vikahistoriatietokantaan lisätään tiedot korjaustoimenpiteistä, jatkossa vastaavien vikojen korjaaminen on nopeampaa ja helpompaa. (Clemm 2009, 135.)

## 4.2 Kokoonpanon hallinta

Verkonhallinnan osalta kokoonpanon hallinta on kenties tärkein osa-alue. Kokoonpanon hallinnalla tarkoitetaan tietoliikenneverkon konfiguroimista käyttökuntoon, konfiguroinnin muuttamista verkon muutosten mukaan ja muutosten dokumentoimista. Väärin konfiguroiduilla laitteilla verkko ei saata toimia ollenkaan, tai vain rajoitetusti. (Ding 2009, 92.)

Konfiguroinnin lisäksi on tiedettävä, mitä laitteita verkkoon on asennettu. On pidettävä huolta, että verkkotopologiakarttaan merkityt laitteet sijaitsevat niissä paikoissa, kuin laitteiden pitäisikin olla. On dokumentoitava laitteiden siirtämiset toiseen paikkaan, ja samalla estettävä ulkopuolisilta omien laitteiden kytkeminen verkkoon aiheuttaen mahdollisia tietoturvahaavoittuvuuksia. (Clemm 2009, 144.)

## 4.3 Käytön hallinta

Käytön hallinnalla tarkoitetaan palveluiden ja verkkoresurssien käytön tallentamista asiakkaan laskutusta varten. Etenkin yritysasiakkaita varten verkon häiriöiden, ja häiriöiden korjausaikojen, tallentaminen on tärkeää palveluntarjoajille. Palveluntarjoajat takaavat tietyn palvelutason, jonka mukaan verkko toimii ennalta määrätyllä tavalla. Jos verkko ei toimi odotetusti, häiriötietokantojen tietojen perusteella voidaan laskea asiakkaalle kuuluvat hyvitykset. (Plevyak & Sahin 2010, 111.)

Laskutuksen lisäksi käytön hallintaan kuuluu käyttöoikeuksien hallinta. On estettävä vihamielisten käyttäjien pääsy verkkoon ilmaisten palveluiden toivossa tai toimiminen toisena käyttäjänä. Palveluiden käyttö ilman maksua aiheuttaa palveluntarjoajalle pelkästään kuluja. Käyttöoikeudettomat ilmaiskäyttäjät hyödyntävät tarjottuja palveluja maksamatta niistä, josta voi seurata esimerkiksi kaistan loppuminen IPTV-tarjoajalta (Intenet Protocol Television). Haitoista joutuvat maksavatkin käyttäjät kärsimään. (Clemm 2009, 154.)

## 4.4 Suorituskyvyn hallinta

Suorituskyvyn hallita vastaa, että tietoliikennejärjestelmän suorituskyky on odotetulla tasolla. Jo suunnitteluvaiheessa verkosta pyritään rakentamaan sellainen, että käyttöönottovaiheessa verkko pystyy takaamaan halutun suorituskyvyn. Kun verkko on otettu käyttöön, verkon suorituskyvyn tilaa tarkkaillaan jatkuvasti ja suorituskyvyn tilasta raportoidaan. Tulosten perusteella voidaan ennustaa mahdollisen suorituskykykapasiteetin heikentymistä ja tehdä päätöksiä verkon päivittämisestä suorituskyvykkäämmäksi. (Verma 2009, 191.)

Verkon kuormituksen ollessa pienempi kuin verkon kapasiteetin määrä verkon suorituskyky on tyypillisesti hyvä. Ajan myötä verkon kuormitus lisääntyy ja verkossa saattaa ilmentyä satunnaisia ja odottamattomia liikennemäärän kasvuja. Tästä seuraa suorituskyvyn selvä heikkeneminen. Heikkenemistä vastaan voidaan taistella suunnittelemalla järjestelmä siten, että todennäköisyys suorituskyvyn heikkenemisille ennalta näkemättömän tapahtuman ilmetessä on pieni. Esimerkik-

si kuormituksen ollessa alle 40 prosenttia todennäköisyys huonoon suorituskykyyn tarkkailtavan ajanjakson aikana on pieni. Jos verkkoa ei voida laajentaa suorituskykyisemmäksi, esimerkiksi taloudellisen tilanteen takia, ja kuormitus lähentelee kapasiteetin rajaa, osalle käyttäjistä voidaan taata hyvä palvelun taso toisten käyttäjien kustannuksella. Tällöin käyttäjät jaetaan eri luokkiin, joille tarjotaan eri tason suorituskykyä. (Verma 2009, 193.)

Verkon suorituskyvyn valvontaan voidaan käyttää useita mittareita. Valvottavilta laitteilta voidaan saada selville suorituskyvyn kannalta hyödyllisiä tietoja, kuten liittymän kautta kulkeneet paketit sekunnin aikana, vasteaika paketin kulkemiseen liitynnästä toiseen, hävinneiden tai vaurioituneiden pakettien määrä tai laitteen prosessorin käyttöaste. (Verma 2009, 195.)

## 4.5 Turvallisuuden hallinta

Verkkoon tunkeutunut luvaton käyttäjä voi saada peruuttamatonta haittaa aikaan. Hän voi esimerkiksi estää ylläpitäjien pääsyn verkkoon, häiritä verkon toimintaa tai haitata laajankin käyttäjäkunnan toimia. Tästä syystä verkon turvallisuuden hallinta on avainasemassa toimivan ja tietoturvallisen verkon hallinnassa. (Verma 2009, 221.)

Tietoturvan hallinta voidaan jakaa viiteen osaan. Autentikoimalla tunnistetaan käyttäjä, joka yrittää kirjautua järjestelmään. Samalla estetään luvattomien käyttäjien pääsy arkaluontoisiin tietoihin. Tietojen eheyden tarkistamisella pyritään varmistamaan järjestelmän tietojen paikkansa pitävyys ja varmistamaan, ettei verkossa oleviin laitteisiin ole asennettuna haittaohjelmia tai vastaavia tietoturvan kannalta haitallisia sovelluksia. Järjestelmään kirjautuneista ja muutoksia tehneistä käyttäjistä on pidettävä kirjaa, jolloin vastuuhenkilöiden henkilöllisyys on helposti selvitettävissä. Näin pyritään jäljittämään mihin kaikkiin järjestelmän osiin luvaton tunkeutuja on päässyt, ja mitä haittaa tunkeutumisesta on seurannut. Mahdollisen palvelunestohyökkäyksen seurauksena ylläpitäjälle on taattava pääsy verkon laitteisiin. Ylläpitäjä voi siten ohjata haitallisen liikenteen muualle ja tarjota verkon käyttäjille pääsyn verkon toimintoihin hyökkäyksestä huolimatta. (Verma 2009, 221 - 222.)

Ulkopuolisten hyökkäysten lisäksi on varauduttava järjestelmän sisältä tuleviin haittoihin. Käyttäjille on sallittava pääsy ainoastaan niihin verkon toimintoihin, joita hän tarvitsee esimerkiksi työnsä puolesta. Salasanat on pidettävä mahdollisimman turvallisina, jotta niiden murtaminen ei olisi helppoa. Salasanat on myös vaihdettava tarpeeksi usein väsytys- ja arvausmenetelmää käyttäviä hyökkäyksiä vastaan. Hallintatiedot, kuten reitittimien konfigurointitiedot, on pidettävä varmuuskopioituna ja turvallisessa paikassa mahdollisen laiterikon varalta. (Clemm 2009, 159.)

Verkon tietoturvahyökkäyksiä vastaan voidaan taistella monin eri tavoin. Tunnetuimpana on palomuuri, joka erottaa kaksi luotettavuustasoltaan erilaista verkkoa. Yleensä palomuuri asetetaan yrityksen sisäverkon ja palveluntarjoajan verkon väliin rajoittamaan liikennettä. Palomuuri voi suodattaa liikennettä monin eri tavoin, kuten IP-osoitteiden tai liikennettä tuottavien ohjelmien mukaan. Tunkeilijan havaitsemisjärjestelmä kerää verkossa kulkevaa liikennettä ja etsii tiettyjä kuvioita verkkoliikenteessä, jotka saattavat olla hyökkäykseen liittyviä. Jos liikenne kulkee salatun yhteyden yli, havaitsemisjärjestelmä ei pysty salauksen vuoksi purkamaan ja tutkimaan pakettien sisältöä. Murron estojärjestelmä estää havaitsemisjärjestelmän havaitseman liikenteen kulun vastaanottajalle. (Verma 2009, 242 - 244.)

Palvelunestohyökkäyksiä vastaan voidaan taistella ottamalla käyttöön käytäntöjä, jotka sallivat tai rajoittavat liikenteen määrää kahden osapuolen välillä. Hyökkääjän yrittäessä siirtää paljon dataa kerralla käytännöt rajoittavat datan määrää pienentäen hyökkäyksen aiheuttamia haittavaikutuksia. Kun hyökkäys on havaittu, on pyrittävä lopettamaan liikenteen kulku luomalla esimerkiksi mustia listoja verkko-osoitteista ja porteista, jotka ovat osallisena hyökkäykseen. Estämällä pääsy osoitteisiin ja portteihin voidaan lopettaa haitallisen liikenteen kulku. Hyökkääjiä varten verkkoon voidaan asettaa ansa. Ansa on verkkoon sijoitettu tunnetulla haavoittuvuudella varustettu laite, joka on itse valvottavasta verkosta erotettu. Tutkimalla ansaan hyökänneen krakkerin aiheuttamia toimia voidaan luoda tunkeilijan havaitsemisjärjestelmälle uusia tunnisteita, joiden avulla voidaan jatkossa havaita vastaavat hyökkäykset. (Clemm 2009, 160 - 161.)

Operatiivisella tasolla verkon ylläpitäjällä on vastuualueita, joista on pidettävä huolta tietoturvallisen ympäristön takaamiseksi. Ulkopuolisilta on estettävä pääsy palvelinhuoneisiin ja laitekaappeihin, jotta voidaan estää fyysisellä tasolla varkaudet, järjestelmien tuhoamiset ja suunnittelemattomat laitteiden sammuttamiset. Odottamattomiin tapahtumiin, kuten tulviin, sähkökatkoksiin ja tulipaloihin on varauduttava jo suunnitteluvaiheessa rakentamalla paloturvallisia tiloja ja varmistettuja sähkönsyöttöjä. Heikoimpana linkkinä fyysisessä turvallisuudessa on vanhojen laitteiden hävittäminen. Laitteiden mukana saattaa kulkea arkaluontoista tai salaista tietoa, kuten palvelimien mukana tietokantoja ja reitittimien mukana konfigurointitietoja paljastaen mahdollisia tietoturvahaavoittuvuuksia yrityksen verkossa. (Verma 2009, 245 - 246.)

Yrityksellä on oltava tietoturvapolitiikka. Tietoturvapolitiikassa määritellä, miten järjestelmä pidetään tietoturvallisena ja häiriöttömänä. Ohjeissa voidaan määritellä, kenellä on pääsy mihinkin järjestelmään, kuinka laitteita konfiguroidaan tietoturvallisiksi, tai vaikka millaisessa muodossa salasanojen on oltava. Ohjeiden avulla ylläpitäjä voi ylläpitää järjestelmää tietoturvallisesti ja yrityksen standardien mukaan. Jos tietoturvapolitiikkaa rikotaan, siihen voidaan puuttua joko automaattisesti tai ylläpitäjän toimesta. Esimerkiksi järjestelmässä on määritelty yrityksen tietoturvapolitiikkaan kuuluva salasanan vähimmäispituus. Jos käyttäjä yrittää syöttää liian lyhyttä salasanaa, järjestelmä pyytää käyttäjää lisäämään merkkejä halutun pituuden saavuttamiseksi. (Verma 2009, 247.)

Järjestelmien auditointi on oleellinen osa järjestelmien turvallisuuden ylläpitämistä. Auditoinnilla tarkoitetaan tilannevedoksen ottamista tietystä järjestelmästä ja vertaamalla tilannevedosta tietoturvapolitiikassa määriteltyihin ohjeisiin. Käytännössä auditointia voi suorittaa yrityksen tietoturvasta vastaava henkilö, joka varmistaa että käytäntöjä seurataan oikein ja laitteet on oikein konfiguroitu. Automaattisesti voidaan esimerkiksi tarkistaa reitittimien konfiguraatioita vertaamalla laitteen asetustiedostoa alkuperäiseen asetustiedostoon. Jos eroja on, syyt muutoksiin pitää selvittää. Tietoturvan testaamista varten yritys voi palkata tietoturva-ammattilaisen yrittämään murtautua yrityksen järjestelmään. Jos murtautuminen onnistuu, verkon ylläpitäjän on paikattava haavoittuvuudet. (Verma 2009, 248.)

# 5 SIVUSTON LUONTI

## 5.1 Johdatus HTML:ään ja CSS:ään

HTML on verkkosivujen teossa käytetty kuvauskieli, jolla määritellään verkkosivun rakenne. HTML:n avulla voidaan määritellä esimerkiksi, mikä osa tekstistä on otsikoita, kuvia tai leipätekstiä. Luomalla linkkejä sivuston eri osien, tai kokonaan muiden sivustojen välillä on yksi HTML:n kantava voima. (Musciano & Kennedy 2006, 8.)

CSS:llä määritellään, miltä HTML:llä luotu verkkosivu näyttää. CSS:n avulla voidaan esimerkiksi määritellä 10 000 verkkosivun ulkonäkö yhdessä ainoassa tyylitiedostossa, jolloin sivustojen ulkonäön muuttaminen on helppoa. Sivuston ylläpitäjälle verkkosivun rungon ja tyylien erossa pitäminen tuo mahdollisuuksia sisällön uudelleen käyttämiseen. CSS:n avulla sivusta voidaan luoda tyylillisesti oma versionsa tulostuskäyttöön tai kustomoida jokaiselle käyttäjäryhmälle erilainen. (Olsson & O'Brien 2008, 3.)

Verkkosivujen luotiin ei tarvitse hankkia kalliita editoreja, vaan sivujen luontiin riittää minimissään tavallinen tekstieditori ja verkkoselain. Työtehokkuuden parantamista varten kannattaa kuitenkin hankkia erityinen editori. Editori voi tarkistaa HTML-koodin eheyden, korostaa kuvaus- ja ohjelmointikielet leipätekstin seasta ja tarjota käsitesanakirjan, jonka avulla voi etsiä tilanteeseen sopivan HTML- tai CSS-tunnisteen. Nettisivua tehdessä on tasaisin väliajoin testattava sivun toimivuus. Sivu testataan suorittamalla verkkosivu selaimessa. Sivua kannattaa testata useammalla erilaisella selaimella, kuten Mozilla Firefoxilla, Internet Explorerilla ja Operalla, koska eri selaimet tukevat eri HTML- ja CSS-elementtejä aiheuttaen sivun näkymisen erilailla eri selaimella. (Musciano & Kennedy 2006, 12 - 14.)

## 5.2 MySQL:n esittely

MySQL on avoimeen lähdekoodiin perustuva relaatiotietokannan hallintaohjelmisto, jonka kehitti suomalainen Michael Widenius vuonna 1995. Viisi vuotta myöhemmin MySQL julkaistiin GNU GPL -lisenssin (GNU General Public License) alaisena, joka mahdollisti MySQL:n käyttämisen ilmaiseksi. (Dyer 2008, 3.)

MySQL sisältää useita tärkeitä ja käyttöä helpottavia ominaisuuksia. Asiakas- / palvelinarkkitehtuuriin perustuvana järjestelmänä MySQL-palvelin kommunikoi useiden eri asiakkaiden kanssa, jolloin tietoa hakevat tai tietokantaan tallentavat asiakkaat voivat sijaita joko samalla tietokoneella kuin palvelin, tai asiakkaat voivat olla fyysisesti eri paikassa kommunikoiden internetin välityksellä. Tuttujen komentojen syöttämistä varten MySQL tukee standardoitua SQL-kieltä, jolloin muiden tietokantaohjelmistojen hallitsemiseen käytetyt komennot toimivat MySQL:ssä. Yhtenä tärkeimmistä ominaisuuksista on käyttöjärjestelmäriippumattomuus. MySQL toimii avoimen lähdekoodinsa ansiosta periaatteessa kaikissa käyttöjärjestelmissä, kuten Linuxissa, Microsoft Windowsissa, Apple Macintosh OS X:ssä ja useissa Unixiin perustuvissa järjestelmissä. Tekstipohjaisen käyttöliittymän lisäksi on mahdollista asentaa esimerkiksi graafinen verkkoselaimessa toimiva käyttöliittymä. (Kofler 2005, 5 - 7.)

## 5.3 Johdatus PHP:hen

PHP on suosittu avoimeen lähdekoodin perustuva palvelinpuolen komentosarja- eli skriptauskieli, jota voidaan käyttää dynaamisten nettisivujen luontiin tai itsenäisten ohjelmien tekoon. PHP:lla tuotetuissa nettisivuissa ohjelmakoodi tulkitaan ennen koodin lähettämistä nettisivulle. Täten nettisivun lähdekoodia tarkastelemalla ei voida sanoa, mitkä osat sivusta on tuotettu PHP:lla. PHP:n avulla voidaan esimerkiksi luoda jokaisella nettisivulla toistuva osio, esimerkiksi sivun alalaidassa näkyvät yrityksen yhteystiedot, tai PHP:n avulla voidaan tallentaa lomakkeeseen syötetyt tiedot MySQL-tietokantaan. (Converse, Park & Morga 2004, 3 - 4, 7.)

PHP toimii Windowsin lisäksi suosituimmissa Unixiin perustuvissa käyttöjärjestelmissä, kuten GNU/Linuxissa ja Mac OS X:ssä. Web-palvelimien tukikin on laaja. Tuettuja palvelimia ovat muun muassa Apache HTTP Server, Microsoft Internet Information Server ja Oracle iPlanet Web Server. (Converse ym. 2004, 11.)

## 5.4 Expectin esittely

Expect on ohjelma, joka kommunikoi tekstipohjaisten interaktiivisten ohjelmien kanssa. Ohjelmalla voidaan luoda skriptejä, joiden avulla Expect osaa syöttää oikeat komennot vasta, kun kohdejärjestelmä vaatii komentojen syöttämistä. Expectiä voidaan tästä syystä käyttää kirjautumiseen ja komentojen antamiseen kohdelaitteelle automaattisesti. (Libes 1994, 1.)

Expect-ohjelmat voidaan kirjoittaa C-, C++- tai Tcl-kielillä. Tcl on tulkattu kieli, jonka päällä Expect toimii tarjoten mahdollisuuden interaktiiviseen toimintaa eri ohjelmien kanssa. (Libes 1994, 3.)

## 5.5 Ubuntu-palvelinkäyttöjärjestelmän esittely

Ubuntu on ilmainen ja vapaaseen lähdekoodiin perustuva GNU/Linux -jakelu ja käyttöjärjestelmä. GNU/Linuxin alkuaikoina GNU/Linux-käyttöjärjestelmän käyttäminen oli vaikeaa. Ei ollut yhtä tiettyä CD:tä tai kasaa diskettejä, joista käyttöjärjestelmän olisi voinut asentaa. Käyttöjärjestelmä piti kasata useista erikseen jaettavista kymmenistä, tai jopa sadoista erillisistä ohjelmista, ja ohjelmien yhdessä toimimaan saamiseen kului paljon tietotaitoa vaativaa aikaa. Asennusvaikeuksista johtuen useimmat GNU/Linuxin käyttäjät olivat alkuaikoina ohjelmoijia. Käyttöjärjestelmälle haluttiin laajempi käyttäjäkunta ja helpompi asennettavuus. Useat projektit kasasivat erikseen jaettuja ohjelmia yhteen, konfiguroivat ohjelmia valmiiksi ja paketoivat ohjelmat helposti asennettavaan mediaan. Projekteja kutsutaan nykyään jakeluiksi. Tunnettuja jakeluita ovat muun muassa Red Hat, Fedora, SUSE, Gentoo ja Debian, johon Ubuntu perustuu. (Hill, Helmke, Graner & Burger. 2011, 13.)

Ubuntu on yksi suosituimmista GNU/Linux-käyttöjärjestelmistä. Syitä suosioon on monia, kuten uusien versioiden tasainen julkaisuväli, käyttäjäystävällinen työpöytä, laaja tuki eri kielille, keskittyminen ainoastaan yhteen ohjelmointikieleen, jolla käyttöjärjestelmää ohjelmoidaan ja laajennetaan sekä laaja yhteistyö ohjelmia luovan yhteisön kanssa. (Hill ym. 2011, 10.)

Ubuntu perustuu muiden jakeluiden tapaan tekstipohjaiseen käyttöliittymään, jonka käyttämisen osaaminen on ensiarvoisen tärkeää. Käyttöliittymä tarjoaa tehokkaan tavan hallita käyttöjärjestelmän kaikkia osia, asentaa uusia ohjelmia ja konfiguroida ohjelmien asetuksia tekstitiedostoja muokkaamalla. Ohjelmien käytön helpottamiseksi jokaisen ohjelman mukana tulee ohjetiedosto. Esimerkiksi Excpect-ohjelman käyttöohjeet saa ruudulle komennolla *man excpect*. Pääkäyttäjä tunnetaan Ubuntussa nimellä Superuser rootin asemesta. Jos halutaan ajaa ohjelmia pääkäyttäjän, on komennon eteen laitettava sana *sudo*. Pääkäyttäjäksi kirjaudutaan komennolla *sudo su*. (Hill ym. 2011, 327, 330 - 331.)

# 6 VALVONTAJÄRJESTELMÄN ASENNUS JA KÄYTTÖÖNOTTO

## 6.1 Lyhyt kuvaus asennuksen vaiheista

Työn tarkoituksena on luoda ilmainen, käyttöjärjestelmä- ja laitteistoriippumaton verkonvalvontasovellus. Sovelluksen avulla verkonvalvonnasta vastaava taho voi yhdellä silmäyksellä nähdä hallittavien laitteiden tilan ja tarpeen vaatiessa pystyy puuttumaan ongelmatilanteisiin. Sovelluksen käyttäminen vaatii ainoastaan nettiselaimella varustetun päätelaitteen, kuten tietokoneen tai matkapuhelimen. Sovellusta pyörittäväksi palvelimeksi kelpaa lähes mikä Linux-yhteensopiva palvelin, koska laitteistovaatimukset ovat hyvin pienet.

Koska valvontasovelluksesta haluttiin tehdä helposti muokattavissa oleva, pääpainona suunnitteluvaiheessa oli käytännönläheisyys. Lähes kaikki, jotka ovat joutuneet konfiguroimaan verkon aktiivilaitteita, ovat käyttäneet työssä komentoliittymää. Komentoliittymää tukevia ohjelmia ovat esimerkiksi Tera Term tai PuTTY. Tästä syystä verkonvalvonnan yhteyskäytännöksi luonnollinen valinta, CLI:n, SNMP:n ja Syslogin väliltä, on yksinkertainen komentoliittymä. Komentoliittymässä komentojen syöttäminen ja vastausten saaminen ovat ihmiselle ymmärrettävässä muodossa, joten syötön ja tulosteen käsitteleminen koneen ymmärtämään muotoon on helppoa. Valvontasovelluksen käyttäjä voi käyttää tuttuja komentoja, kuten *show ip interface FastEthernet1*, muuttaessaan valvontasovelluksen toimintaa.

Tietojen hakemisesta valvottavalta laitteelta vastaa Expect-niminen ohjelma. Expect on käytännössä ainoa saatavilla oleva ohjelma, jonka avulla valvontajärjestelmä ja valvottava laite voivat keskustella interaktiivisesti komentoliittymän kautta.

Valvottavien laitteiden merkillä on tässä työssä väliä. Valvontasovelluksella voidaan valvoa ainoastaan Cisco-merkkisiä laitteita, joissa on käyttöjärjestelmänä Cisco IOS. Käyttöjärjestelmän versiolla ei pitäisi suuremmin väliä, mutta ainakin versio 12.4:llä valvontajärjestelmä toimii. Muiden valmistajien laitteita voidaan lisätä tiedostoja muuttamalla. Jos valvottavan laitteen merkkiä ei oteta huomioon,

niin ainoana vaatimuksena laitteelle on, että laitetta voidaan konfiguroida tekstipohjaisesti joko telnet- tai SSH-yhteyden kautta. Työhön sisältyy Cisco IOS -yhteensopiva ohjelmakoodi, jolla laitteiden lisääminen valvontaan onnistuu helposti.

Työn keskeisenä osapuolena on Linux-palvelin, johon on asennettuna Ubuntu. Ubuntun valintaan päädyttiin, koska kyseinen käyttöjärjestelmä pohjautuu vapaaseen lähdekoodiin ja on samalla yksi käytetyimmistä Linux-jakeluista. Ongelmatilanteissa yhteisöjen verkkosivuilta löytyy runsaasti ratkaisuja. Ubuntun asemesta käyttöjärjestelmäksi kelpaa muukin Linux-jakelu, kuten palvelinkäytössä yleinen CentOS. Käyttöjärjestelmään asennetaan useita laajennuksia, joita valvontasovellus tarvitsee. Expect-työkalulla kirjaudutaan valvottavaan laitteeseen sisään, haetaan valvontatiedot ja tallennetaan tiedot tekstitiedostoon. Expect käyttää laitteeseen kirjautumiseen joko telnet- tai SSH-protokollaa.

PHP-ohjelmointikielellä (PHP: Hypertext Prepocessor) jäsennetään tekstitiedostosta vaaditut tiedot MySQL-tietokantaan, jossa tiedot ovat selvästi jäsennettyinä. PHP:n valintaa ohjelmointikieliksi puolustavat monet seikat. PHP:llä voidaan luoda interaktiivisia verkkosivuja, käsitellä tiedostoja ja kaiken lisäksi PHP tukee natiivisti MySQL-komentoja. MySQL valittiin tietokantaohjelmistoksi suosion, PHP-yhteensopivuuden ja ohjelmistolisenssinsä ansiosta.

Ohjelman käyttäjä näkee ainoastaan verkkosivun, johon on listattuna valvottavat laitteet ja laitteiden tiedot. Verkkosivun perusta on tehty HTML:llä (Hypertext Markup Language) ja ulkoasu muokattu CSS:llä (Cascading Style Sheets). Valvottavien laitteiden tiedot haetaan sivulle PHP:llä MySQL-tietokannasta.

Valvontasovellus voidaan asettaa hakemaan valvottavien laitteiden tiedot automaattisesti tasaisin väliajoin. Tämä suoritetaan cron-ajastuspalvelulla. Valvontasivua voidaan myös päivittää automaattisesti lisäämällä verkkosivulle pari riviä JavaScriptiä, joka huolehtii sivun uudelleen lataamisesta tietyn väliajoin.

## 6.2 Ubuntun asennus

Käyttöjärjestelmäksi valittiin Ubuntu 10.04 LTS Desktop, joka perustuu Debian-Linux-jakeluun. Työn tarkoituksena on rakentaa kustannustehokas ja tietoturvallinen käyttöympäristö, jolloin Linux on luonnollinen valinta maksuttomuutensa ja tietoturvansa takia. Ubuntu puolestaan on työn kirjoitushetkellä suosituin Linux-jakelu Distrowatch.com:n (http://distrowatch.com/) mukaan. 10.04 LTS on viimeisin pitkäaikaisen tuen (Long Term Support) takaava Ubuntun versio. Desktop- eli työpöytäversion valintaa tukee helppokäyttöisyys. Version mukana tulee graafinen käyttöliittymä, jonka avulla asetusten muuttaminen on helpompaa tekstipohjaista käyttöliittymää vierastavalle.

Palvelimen pohjana voidaan käyttää muutakin kuin Ubuntu 10.04 LTS Desktop -käyttöjärjestelmää. Valinta voidaan tehdä omien mieltymysten mukaan, koska työssä käytetyt ohjelmat pitäisi toimia muissakin Linux-jakeluissa.

Palvelimen asentaminen aloitetaan lataamalla käyttöjärjestelmän levykuvatiedosto kehittäjien kotisivuilta. Työssä käytetty Ubuntu on saatavilla osoitteessa https://www.ubuntu.com. Ladattu levykuvatiedosto voidaan joko polttaa CD-levylle ja asentaa palvelimelle tai käyttää käyttöjärjestelmää virtuaalisena. Virtualisointiin on saatavilla useita maksuttomia tai maksullisia ohjelmia. Yhtenä esimerkkinä voidaan antaa maksuton VMware Server. Itse käyttöjärjestelmän asennus on helppoa graafisen käyttöliittymän avulla, joten siihen ei sen suuremmin puututa.

## 6.3 Ubuntu LAMP -palvelimen asennus ja käyttöönotto

LAMP (Linux, Apache HTTP Server, MySQL, Perl/PHP/Python) asentaa lyhyesti sanottuna palvelimelle tarvittavat ohjelmat tietokannalla varustettujen dynaamisten nettisivujen tarjoamiseen. LAMP asennettaan palvelimelle terminaalissa annettavalla komennolla *sudo apt-get install lamp-server^*. On huomioitava, että komennon viimeinen merkki on sirkumfleksi. Asennuksen aikana pyydetään syöttämään MySQL-palvelimen pää- eli root-käyttäjälle uusi salasana. Jotta PHP saadaan toimimaan Apachen kanssa, Apache on käynnistettävä uudelleen. Terminaa-

lissa annetaan komento *sudo /etc/init.d/apache2 restart*. Jotta php-skriptejä voidaan suorittaa komentoliittymästä, on asennettava php5-cli-paketti komennolla *sudo apt-get install php5-cli*.

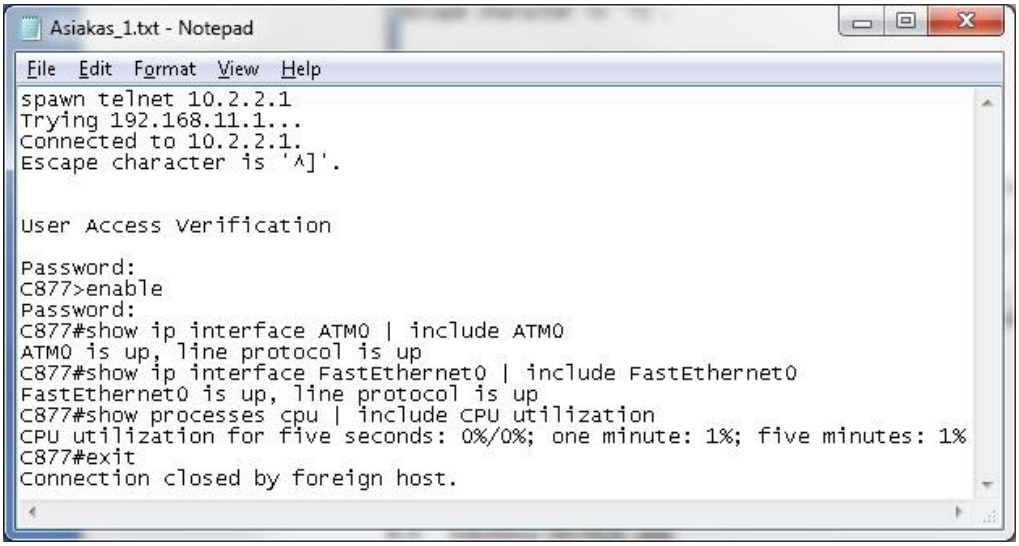
PHP:n ja Apachen toimivuuden varmistamiseksi /var/www-kansioon tehdään PHP-testitiedosto. Komennolla *sudo pico /var/www/testi.php* avataan tekstieditori. Editoriin kirjoitetaan *<?php echo "PHP toimii."; ?>* , tallennetaan ja poistutaan editorista. Avataan www-selain, esimerkiksi Mozilla Firefox, ja syötetään osoiteriville osoite http://localhost/. Jos ensimmäisellä rivillä lukee "It works!", Apache on toimintakunnossa. PHP:n toiminnan varmistamiseksi osoiteriville kirjoitetaan osoitteeksi http://localhost/testi.php. Jos ensimmäisellä rivillä lukee "PHP toimii!", niin PHP on toimintakunnossa.

### 6.3.1 Expectin asennus ja käyttö kirjautumisessa valvottavaan laitteeseen

Työssä Expectiä käytetään kirjautumaan valvottavaan laitteeseen, hakemaan halutut tiedot ja tallentamaan tiedot tekstitiedostoihin jatkokäsittelyä varten. Expect ei tule Ubuntun mukana, joten Expect on itse asennettava. Expect asennetaan terminaalissa annettavalla komennolla *sudo apt-get install expect*. Ohjelma ei vaadi asetusten säätämistä, vaan ohjelmaa voi käyttää suoraan asentamisen jälkeen.

Liitteessä 1 on tietojen_haku.exc-niminen Expect-skripti, jolla kirjaudutaan Cisco-merkkiseen reitittimeen, kytkimeen tai vastaavaan verkossa toimivaan laitteeseen. Skriptillä haetaan tarvittavat tiedot, eli ulko- ja sisäliittymien fyysiset ja linja-protokollan tilat sekä prosessorin käyttöasteen viiden viimeisen minuutin aikana. Ohessa selitetään liitteen 1 oleellisimmat komennot. Komennolla *set*, esimerkiksi *set password "cisco"*, luodaan password-niminen muuttuja, jonka sisältö on merkkijono *cisco*. Skriptissä muuttujiin viitataan dollarimerkillä ($). Skriptin alussa on määritelty viisi muuttujaa: host (valvottavan laitteen IP-osoite) , password (reitittimen salasana), enablePassword (enable-tilan salasana), wanInt (ulkoverkon liittymän tunnus) ja lanInt (sisäverkon liittymän tunnus). Liittymien tunnukset ovat merkistökokoriippuvaisia, eli esimerkiksi fastethernet0 ja FastEthernet0 eivät ole vastaavia. Liittymän oikean kirjoitusasun saa selville Cisco-laitteissa esimerkiksi komennolla *show ip interfaces brief*.

Liitteen 1 skriptissä *spawn telnet $host* -komennolla otetaan telnet-yhteys valvottavaan laitteeseen. Komennolla *expect "Password:"* odotetaan vastapään laitteen lähettävän merkkijonon *Password:*. Heti, kun vastapään laite on lähettänyt kyseisen merkkijonon, Expect-skripti suorittaa komennon *send "$password\n". $password* viittaa skriptin alussa määriteltyyn muuttujaan, joka siis on kirjautumisen käytettävä salasana ja merkkijono *\n* tarkoittaa newline-merkkiä eli rivinvaihtoa. Rivinvaihto toimii kuten näppäimistön enter-näppäin, eli rivinvaihdon avulla skripti lähettää salasanan valvottavalle laitteelle. Skriptin lopussa oleva *expect eof* odottaa telnet- (tai ssh-) prosessin päättymistä, eli EOF-merkkiä (End-of-file), jonka jälkeen skriptin suoritus päättyy. Suorituksen tuloksena saadaan txt-päätteinen tekstitiedosto, jonka nimi on haettu skriptin $LaiteID-muuttujasta, esimerkiksi tiedoston nimi voi olla Operaattori_1.txt. Tiedosto sisältää koko skriptin aikaisen keskustelun valvottavan laitteen ja skriptiä suorittaneen palvelimen välillä. Kuviossa 5 on skriptin luoma esimerkkitiedosto.




KUVIO 5. Tietojen_haku.exc-skriptin luoma tekstitiedosto

### 6.3.2 MySQL-tietokannan luonti

Reitittimistä saatavat tiedot tallennettaan MySQL-tietokantaan. Taulukko 2 on luotavan tietokantataulun, nimeltään Valvonta, rakenne. Taulu koostuu seitsemästä sarakkeesta, joista LaiteID- ja IP-sarakkeiden solut ovat kiinteäarvoisia ja ainoita käsin syötettäviä arvoja. LaiteID-sarakkeen arvoilla tunnistetaan, mikä laite on kyseessä. Taulukko 3 on esimerkkitietokantaulu, johon puuttuvat tiedot on haettu automaattisesti.

- LaiteID määrää laitteen ennalta määrätyn yksilöllisen tunnisteen. Tässä tapauksessa se on reitittimen isäntänimi, englanniksi hostname. Solun arvo syötetään käsin. Arvo ei tule jatkossa muuttumaan, ellei sitä muuteta uudelleen käsin. Työssä käytettävä arvo 'Operaattori' tarkoittaa operaattorin reititintä ja 'Asiakas' asiakkaan reititintä.
- IP-sarakkeeseen syötetään reitittimen ulkoverkon IP-osoite, esimerkiksi 192.168.1.2.
- Wan_stat-sarakkeeseen haetaan reitittimen ulkoliittymän fyysinen tila. Mahdolliset tilat ovat up, down ja administratively down.
- Wan_prot-sarakkeeseen haetaan reitittimen ulkoliittymässä toimivan linjaprotokollan tila. Mahdolliset tilat ovat up ja down.
- Lan_stat-sarakkeeseen haetaan reitittimen sisäliittymän fyysinen tila. Mahdolliset tilat ova up, down ja administratively down.
- Lan_prot-sarakkeeseen haetaan reitittimen sisäliittymässä toimivan linjaprotokollan tila. Mahdolliset tilat ovat up ja down.
- CPU-sarakkeeseen haetaan reitittimen keskimääräinen prosessorinkulutus prosentteina viimeisen viiden minuutin aikana.
- Paivitetty-sarakkeeseen haetaan automaattisesti päiväys, kun yhtäkin rivin soluista päivitetään.

TAULUKKO 2. MySQL-tietokantaan luotava Laiteet-niminen tietokantataulu

| LaiteID | IP | Wan_stat | Wan_prot | Lan_stat | Lan_prot | CPU | Paivitetty |
|---|---|---|---|---|---|---|---|
| Operaattori_1 | 10.1.1.2 | | | | | | |
| Asiakas_1 | 10.2.2.1 | | | | | | |

TAULUKKO 3. Esimerkkinä käytössä oleva MySQL-tietokantataulu

| LaiteID | IP | Wan_stat | Wan_prot | Lan_stat | Lan_prot | CPU | Paivitetty |
|---|---|---|---|---|---|---|---|
| Operaattori_1 | 10.1.1.2 | up | up | up | up | 82 | 2011-08-15 20:58:27 |
| Asiakas_1 | 10.2.2.1 | up | up | up | down | 5 | 2011-08-14 21:03:05 |

MySQL:ään luodaan uusi tietokanta ja tietokantaan pääsevä erillinen käyttäjä. MySQL:n käynnistämiseksi terminaalissa syötetään komento *mysql -u root -p* ja syötetään MySQL:n pääkäyttäjän salasana. Uusi tietokanta, nimellä Valvonta, luodaan komennolla *create database Valvonta;*. Seuraavaksi luodaan uusi käyttäjä tunnukseltaan 'valvomo' ja salasanaltaan 'passu' komennolla *CREATE USER 'valvomo'@'localhost' IDENTIFIED BY 'passu';*. Valvonta-tietokantaan annetaan valvomo-käyttälle pilkulla erotetut oikeudet komennolla *grant CREATE,INSERT,DELETE,UPDATE,SELECT,ALTER ON Valvonta.* TO 'valvomo'@'localhost';*. Lopuksi annetaan komento *FLUSH PRIVILEGES;*, joka lataa käyttäjän käyttöoikeudet uudelleen. MySQL:stä poistutaan komennolla *EXIT;*.

Juuri luodulla käyttäjällä *valvomo* ja salasanalla *passu* kirjaudutaan MySQL:ään komennolla *mysql -u valvomo -p,* painamalla enter ja syöttämällä salasanan *passu.* Luodaan taulukon 2 mukainen taulu MySQL-tietokantaan. Ensin valitaan tietokanta nimeltä Valvonta komennolla *use Valvonta;*. Laitteet-niminen taulu luodaan komennolla

```
CREATE TABLE Laitteet
(
LaiteID VARCHAR(25) NOT NULL,
IP VARCHAR(25),
Wan_stat VARCHAR(25),
Wan_prot VARCHAR(25),
Lan_stat VARCHAR(25),
Lan_prot VARCHAR(25),
CPU VARCHAR(25),
Paivitetty TIMESTAMP ON UPDATE CURRENT_TIMESTAMP NOT NULL
DEFAULT CURRENT_TIMESTAMP;
)
```

### 6.3.3 Sivuston rungon luominen HTML:llä ja tyylien CSS:llä

Sivuston rungon, ja käyttäjälle näkyvän käyttöliittymän, muodostavat liitteen 2 valvonta.php-niminen HTML-dokumentti ja liitteen 3 tyylit.css-niminen tyylitiedosto. HTML-dokumentin tarkoitus on luoda verkkoselaimelle tulkittava verkkosivu, ja tyylitiedoston tarkoitus on muokata HTML-dokumentin tyylejä.

Valvonta.php on hyvin yksinkertainen tiedosto. Tiedosto sisältää lähinnä pakolliset HTML-dokumentin määrittelyt, linkit tyylitiedostoon ja itse sisällöstä vastaavaan mysql-esitys.php-tiedostoon. Tyylit.css-tyylitiedoston avulla muokataan sivua käyttäjäystävällisempään ja esteettisempään suuntaan. Ilman tyylitiedostoa HTML-dokumentti on varsin karun näköinen, mutta muutamilla pienillä määrittelyillä voidaan muuttaa värejä, taulukon reunoja ja luoda huomattavasti luettavampi taulukko.

### 6.3.4 Oleellisten tietojen hakeminen PHP:lla ja tietojen tallentaminen MySQL-tietokantaan

Expect-skriptin luomasta tekstitiedostosta on saatava haettua oleelliset tiedot, ja tallennettava tiedot MySQL-tietokantaan. Tiedoston parsiminen ja tallentaminen hoidetaan liitteessä 6 olevalla mysql-tallennus.php-nimisellä tiedostolla.

Ennen varsinaisen ohjelmakoodin toiminnan tarkastelemista on huomioitava vakiomuuttujat.php-tiedosto, joka on liitteenä 4. Kyseinen tiedosto sisältää yleisiä vakiomuuttujia, kuten MySQL-palvelimen osoitteen, käyttäjätunnuksen, salasanan sekä valvottavan laitteen uniikin tunnuksen ja verkkoliittymien tunnukset. Vakiomuuttujat.php-tiedoston tarkoitus on kerätä kaikki vakiomuuttujat yhteen tiedostoon, jota muut samoja muuttujia tarvitsevat tiedostot voivat käyttää. Keräämällä tiedot yhteen paikkaan voidaan esimerkiksi palvelimen osoitteen muuttuessa muuttaa laitteen osoitemuuttuja yhteen tiedostoon sen sijaan, että osoite syötettäisiin kaikkiin osoitetta vaativiin tiedostoihin. Näin säästetään aikaa ja minimoidaan suoritettavaan ohjelmakoodiin mahdolliset virheellisestä syötöstä aiheutuneet häiriöt.

Mysql-tallennus.php-tiedosto jakautuu kahteen funktioon. TiedotMuuttujiin-funktio vastaa tietojen_haku.exc:n luoman tekstitiedoston jäsentämisestä ja haluttujen tietojen syöttämisestä muuttujiin. Funktio lukee tekstitiedostoa yhden rivin kerrallaan ja tallentaa jokaisen sanan, joka on erotettu välilyönnillä, taulukkoon. Silmukka vertaa taulukon muuttujia tiedettyihin arvoihin ja osuman sattuessa tallentaa halutun tiedon muuttujaan. Esimerkiksi taulun sisältö on seuraava: "FastEthernet0 is up, line protocol is up", jossa yksi muuttuja vastaa yhtä sanaa. Silmukka etsii peräkkäin tulevia sanoja "FastEthernet0" ja "is". Kun halutut sanat esiintyvät, sanoja seuraava arvo, tässä tapauksessa "up", tallennetaan muuttujaan. Samalla tavoin haetaan ja tallennetaan kaikki halutut tiedot muuttujiin. Jos yhteyttä , tekstitiedostosta saatavan tiedon mukaan, ei ole saatu valvottavaan laitteeseen, kaikille muuttujille annetaan arvo "tuntematon".

TiedotTietokantaan-funktio vastaa edellisen funktion muuttujien arvojen syöttämisestä tietokantaan. Funktio luo yhteyden tietokantapalvelimeen sekä muodostaa ja syöttää MySQL-palvelimelle lauseita. Lauseet päivittävät laitteiden tilaa vastaavat muuttujien arvot tietokantaan. Funktion syöttämä lause on tyypillisesti seuraavanlainen:
"UPDATE Laiteet SET Wan_stat='up' WHERE LaiteID='Operaattori_1'".

### 6.3.5 Tietojen hakeminen tietokannasta valvontasivulle

Tietojen hakemisesta MySQL-tietokannasta vastaa liitteessä 5 oleva mysql-esitys.php-niminen tiedosto. Käyttäjälle näkyvä HTML-dokumentti, valvonta.php, kutsuu mysql-esitys.php-tiedostoa, joka tuottaa käyttäjälle näkyvän taulukon. Taulukosta selviää valvottavien laitteiden tila.

Ohjelmistokoodin suoritus aloitetaan ottamalla ensin yhteys MySQL-palvelimeen. Koko tietokantataulu haetaan muistiin, josta tulostetaan verkkosivulla näkyvän taulukon otsikoksi tietokannan nimi ja käytettävä taulu. Taulukon tulostamista jatketaan hakemalla tietokannasta sarakkeiden otsikot, joiden lisäksi luodaan yksi uusi sarake nimeltä Tilanne.

Tilanne-sarakkeesta näkee yhdellä vilkaisulla laitteen tilan. Vakavuustasoja on neljä kappaletta, ja tasoja kuvataan eri värisillä neliökulmioilla laitteen tietojen perässä: tuntematon (harmaa), ok (vihreä), varoitus (keltainen) ja hälytys (punainen). Tasojen määrittelyt menevät seuraavasti:

- tuntematon: valvottavaan laitteeseen ei ole saatu yhteyttä.
- ok: kaikki verkkoliittymät ovat ylhäällä, ja prosessorin käyttöaste viimeisen viiden minuutin aikana on alle liitteen 4 vakiomuuttujat.php:ssä määritellyn arvon.
- varoitus: sisäverkon liittymän fyysisen ja linjaprotokollan tila on "down" ja prosessorin käyttöaste viimeisen viiden minuutin aikana on alle määritellyn arvon.
- hälytys: kun yksikin seuraavista ehdoista täyttyy: ulkoverkon liittymän fyysinen tila on "down", ulkoverkon liittymän linjaprotokollan tila on "down" tai "administratively down", sisäverkon liittymän tila on "administratively down" tai prosessorin käyttöaste viimeisen viiden minuutin aikana on sama tai suurempi kuin määritelty arvo.

Vakavuustasojen määrittelyihin käytetään liitteen 4 vakiomuuttujat.php-tiedostoa. Tiedostossa laitteen eri tiloille, kuten sisäverkon liittymän fyysiselle tilalle "down" voidaan antaa tietty määrä virhepisteitä. Mitä enemmän virhepisteitä annetaan, sitä suurempi on vakavuustaso. Vakavuustasojen nostoon vaadittavaa vir-

hepisteiden määrääkin voidaan muuttaa vakiomuuttujat.php-tiedostoa muokkaamalla. Muokattavuuden ansiosta vakavuustasoja voidaan määritellä eri laitteille erikseen ja omaan käyttöön sopivaksi. Jos esimerkiksi halutaan, että järjestelmä antaa kriittisen hälytyksen prosessorin käyttöasteen noustessa liian korkealle, muutetaan laitteen vakiomuuttujat.php-tiedoston muuttujia $CPUyliRajan ja $halytystaso. Muuttuja $halytystaso määrää, minkä verran virhepisteitä pitää vähintään olla, jotta kriittinen hälytys annetaan. Muuttujan $CPUyliRajan arvo sisältää virhepisteet, jotka otetaan vakavuustason laskennassa huomioon, jos prosessorin käyttöaste on liian korkea. Jos kummankin muuttujan arvoiksi asetetaan yhtä suuri luku, esimerkiksi luku kolme, kriittinen hälytys annetaan prosessorin käyttöasteen noustessa liian korkealle.

Vakavuustason määrittelyn jälkeen verkkosivun taulukkoon tulostetaan jokaisesta laitteesta tietokannasta haetut tiedot ja vakavuustaso. Jokaisen laitteen IP-osoitekohta muutetaan klikattavaksi linkiksi. Klikkamalla linkkiä avataan telnet-yhteys laitteeseen hallinta varten. Yhteyden muodostamiseksi verkkoselaimen asetuksista on määriteltävä sopiva pääteohjelma, kuten Nutty, avaamaan telnet-protokollaa olevat linkit. Koodia muokkaamalla telnet-linkit ovat helposti muutettavissa SSH-linkeiksi.

## 6.4 Hakemistorakenne ja laitteen lisääminen järjestelmään

Työn koejärjestelyn hakemistorakenne oli seuraava. /Var/www-hakemistoon luotiin uusi hakemisto nimeltä laitteet. Laitteet-hakemistoon siirrettiin mysql-esitys.php-, mysql-tallennus.php-, tietojen-haku.exc, tyylit.css-, vakiomuuttujat.php- ja valvonta.php-tiedostot. Siirretyille ja uusille tiedostoille on muistettava laittaa käyttöoikeudet kuntoon chmod-komennolla. Toimivuuden testauksen kannalta kaikkiin tiedostoihin voidaan asettaa kaikille käyttäjille kaikki oikeudet, mutta tietoturvan kannalta tämä ei ole viisasta. Tässä esimerkissä lisäämme uuden laitteen järjestelmään. Laitteen yksilöllinen tunnus on Asiakas_2, IP-osoite 10.1.1.3, salasana "cisco", enable-tilan salasana "cisco", ulkoverkon liittymän tunnus FastEthernet0/1 ja sisäverkon liittymän tunnus FastEthernet0/24.

Ensin lisätään laite tietokantaan kirjautumalla MySQL-palvelimelle. MySQL:ssä komennolla *use Valvonta;* valitaan Valvonta-niminen tietokanta. Uusi laite lisätään Laitteet-tietokantaan komennolla
*INSERT INTO Laitteet (LaiteID, IP) VALUES ('Asiakas_2', '10.1.1.3');*. Tietokannasta poistutaan komennolla *EXIT;*.

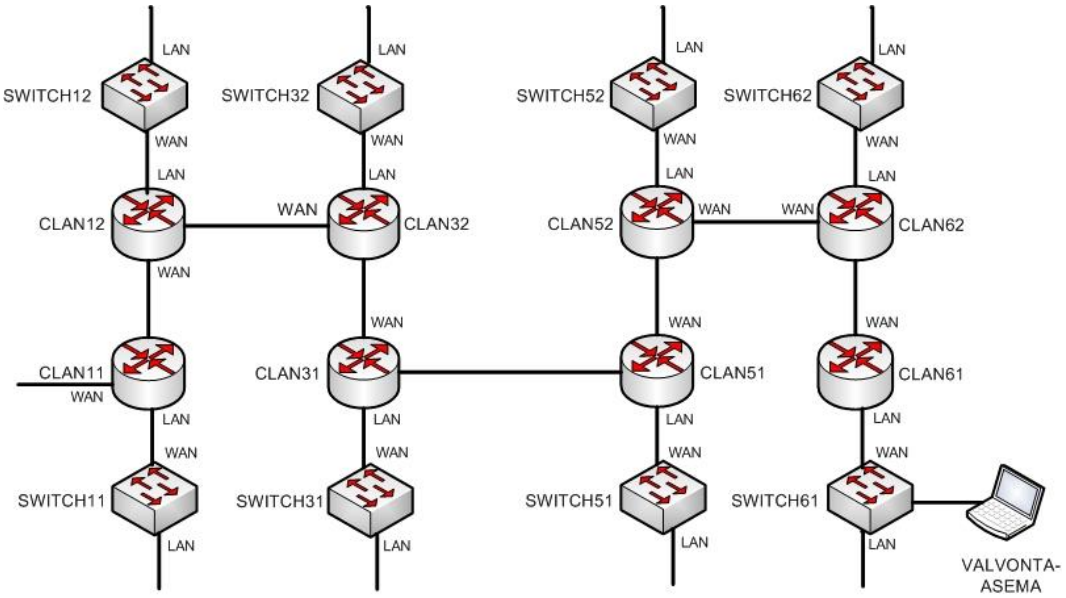
Kun laite on lisätty tietokantaan, luodaan /var/www/laitteet-kansioon uusi kansio, jonka nimi on sama kuin valvottavan laitteen, eli Asiakas_2. Kansio luodaan komennolla *mkdir Asiakas_2.* Juuri luotuun kansioon siirretään tietojen_haku.exc-, mysql-tallennus.php ja vakiomuuttujat.php-tiedostot komennolla *cp tietojen_haku.exc mysql-tallennus vakiomuuttujat.php Asiakas_2/.* Siirrytään Asiakas_2-kansioon ja muokataan ensin tietojen_haku.exc-tiedostoa, joka suorittaa valvottavaan laitteeseen kirjautumisen ja tietojen haun. Muokattavassa tiedostossa on kuusi muuttujaa, joihin on asetettava oikeat arvot. LaiteID-muuttujan arvoksi asetetaan Asiakas_2, host-muuttujan arvoksi 10.1.1.3, password- ja enablePassword-muutujiksi cisco, wanID-muuttujaksi FastEthernet0/1 ja lanID-muuttujaksi FastEthernet0/24. Tiedosto tallennetaan ja suljetaan.

Avataan työkansiossa oleva tiedosto vakiomuuttujat.php. Tarpeen vaatiessa voidaan muuttaa MySQL-palvelimen muuttujia ja virhemuuttujia, mutta tässä esimerkissä oletetaan, että kyseiset muuttujat on jo määritelty ennen kuin vakiomuuttujat.php-tiedosto on kopioitu Asiakas_2-kansioon. Nyt muutetaan ainoastaan laitetietojen muuttujat, joita on kolme kappaletta. $LaiteID-muuttujan arvoksi asetetaan Asiakas_2, $wanID-muuttujaksi FastEthernet0/1 ja $lanID-muuttujaksi FastEthernet0/24. Tallennetaan tiedosto ja suljetaan.

Nyt tarpeelliset muokkaukset on tehty uuden laitteen lisäämiseksi. Tiedot valvottavasta laitteesta haetaan valvottavan laitteen työkansiossa annettavalla komennolla *./tietojen_haku.exc.* Haetut tiedot tallennetaan tietokantaan komennolla *php mysql_tallennus.php.* Valvontasivulle, joka siis esittää valvottavien laitteiden tilan, pääsee verkkoselaimella syöttämällä osoitteeksi
http://192.168.11.10/laitteet/valvonta.php. Osoite 192.168.11.10 viittaa palvelimeen, jolla valvontajärjestelmää ajetaan.

# 7 VALVONTAJÄRJESTELMÄN TESTAUS

Valvontajärjestelmän testaus suoritettiin Lahden ammattikorkeakoulun tekniikan alan tietoverkkolaboratoriossa. Valvottaviksi laitteiksi valittiin kahdeksan Cisco-merkkistä reititintä ja kahdeksan samanmerkkistä kytkintä. Kytkimien perään oli kytketty tietokoneet, jotka eivät kuuluneet valvottaviin laitteisiin. Laitteet oli valmiiksi liitetty laboratorion CiscoNET2-verkkoon. Valvontajärjestelmä on tarkoitettu sijoitettavaksi tähtimäisen verkkotopologian keskukseen, jossa palveluntarjoaja voi valvoa loppukäyttäjien verkon päätepisteissä sijaitsevia laitteita. Testissä käytetty verkko oli kuitenkin topologialtaan väylämäinen, joten verkko ei aivan vastannut malliltaan lopullista topologiaa. Testin kannalta verkon rakenteella ei ollut merkitystä, koska kokeen tarkoituksena oli kokeilla valvontajärjestelmän liittämistä verkkoon, toimintaa normaali- ja vikatilanteissa sekä vikatilanteista toipumista.




KUVIO 6. Testissä käytetyn verkon rakenne

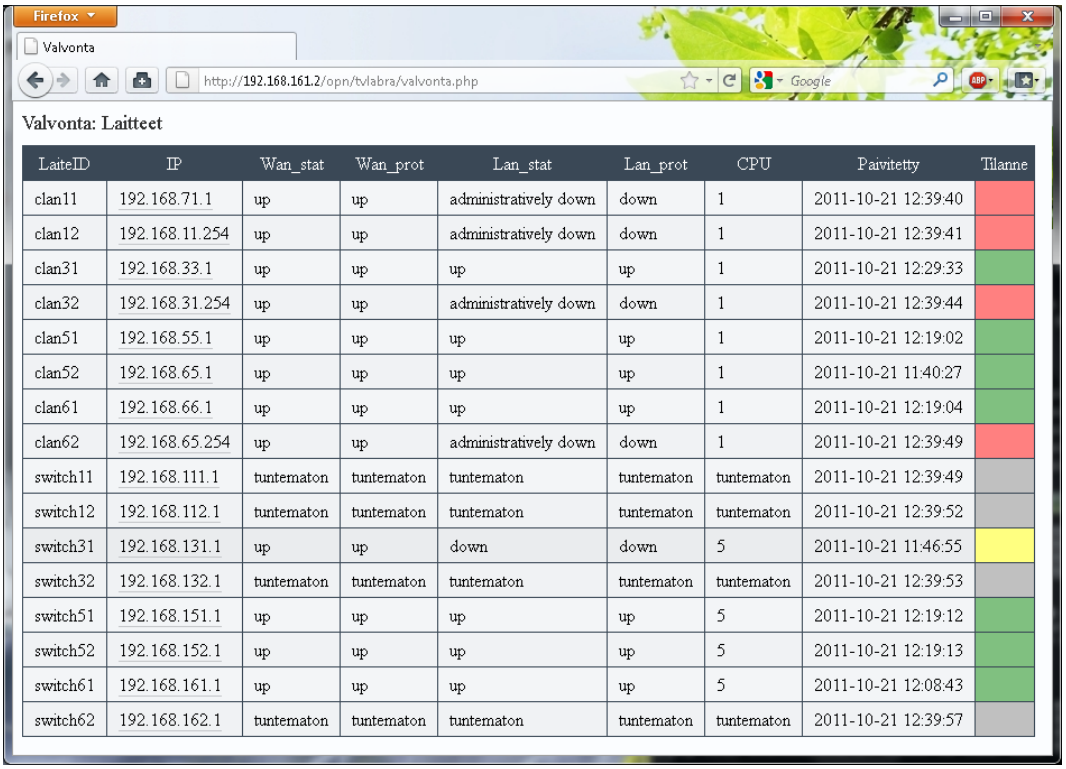
Verkon rakenne on kuvattu kuviossa 6. Kuviosta selviää, että jokaiselle reitittimille ja kytkimelle on annettu yksilöllinen nimi, ja jokaiselle valvottavalle laitteelle on määritelty yksi WAN- eli ulkoverkon liittymä ja yksi LAN- eli sisäverkon liit-

tymä. Verkon laitteet konfiguroitiin perusasetuksin, eli liittymille annettiin IP-osoitteet, asetettiin salasanat ja laitteiden yksilölliset tunnukset sekä otettiin käyttöön RIPv2-reititysprotokolla.

Valvonta-asemana toimi muutaman vuoden ikäinen kannettava tietokone, johon oli asennettuna virtuaalikoneiden ajamiseen tarkoitettu VMware Workstation. Workstationilla ajettiin virtuaalikoneena olevaa valvontajärjestelmää. Valvontajärjestelmälle annettiin kiinteä IP-osoite, joka kuului samaan verkkoon CLAN61-reitittimen LAN-liittymän kanssa, ja lisättiin jokainen valvottava laite valvontajärjestelmään. Tietojen hakua (eli tietojen_haku.exc-tiedostojen ajamista) ja tietojen tallentamista (eli mysql-tallennus.php-tiedostojen ajamista) varten luotiin yksinkertainen skripti, jonka avulla valvottavista laitteista haettiin halutut tiedot, ja tallennettiin tiedot tietokantaan laite kerrallaan automaattisesti. Skriptiä ajettiin aina valvontatilanteen, kuten linkin katkeamisen, muuttuessa, ja tuloksia seurattiin avaamalla valvonta.php-tiedosto verkkoselaimella.

## 7.1 Testin tulokset

Testiympäristössä ajettiin useita erilaisia testejä. Laitteiden liittymiä sammutettiin valvottavan laitteen hallinnasta ja verkkokaapeleita irrotettiin fyysisesti laitteiden liittimistä. Samalla ajettiin skriptiä, jonka avulla valvottavista laitteista saatiin yhteyden onnistuessa hakemaan ja tallentamaan tiedot. Kun laitteeseen saatiin yhteys, tietojen hakuun ja tallentamiseen meni muutama sekunti laitetta kohden. Jos yhteyttä ei saatu, tietojen_haku.exc:ssä on määritelty aika, tässä tapauksessa 30 sekuntia, jonka jälkeen päätellään laitteen olevan saavuttamattomissa. Tällöin laitteen tilojen arvoiksi saatiin arvo "tuntematon".

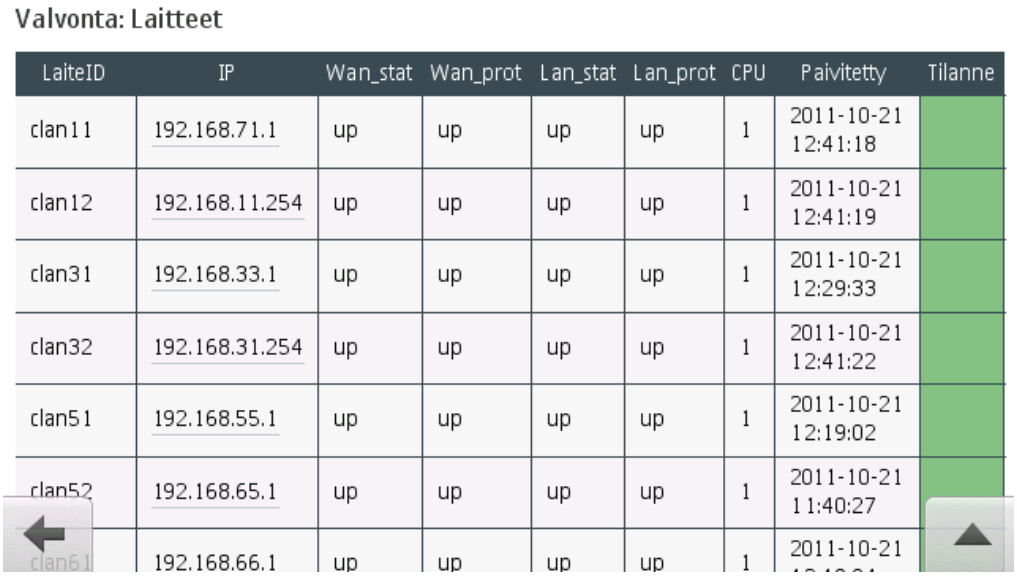



Valvonta: Laitteet

| LaiteID | IP | Wan_stat | Wan_prot | Lan_stat | Lan_prot | CPU | Paivitetty | Tilanne |
|---|---|---|---|---|---|---|---|---|
| clan11 | 192.168.71.1 | up | up | administratively down | down | 1 | 2011-10-21 12:39:40 | |
| clan12 | 192.168.11.254 | up | up | administratively down | down | 1 | 2011-10-21 12:39:41 | |
| clan31 | 192.168.33.1 | up | up | up | up | 1 | 2011-10-21 12:29:33 | |
| clan32 | 192.168.31.254 | up | up | administratively down | down | 1 | 2011-10-21 12:39:44 | |
| clan51 | 192.168.55.1 | up | up | up | up | 1 | 2011-10-21 12:19:02 | |
| clan52 | 192.168.65.1 | up | up | up | up | 1 | 2011-10-21 11:40:27 | |
| clan61 | 192.168.66.1 | up | up | up | up | 1 | 2011-10-21 12:19:04 | |
| clan62 | 192.168.65.254 | up | up | administratively down | down | 1 | 2011-10-21 12:39:49 | |
| switch11 | 192.168.111.1 | tuntematon | tuntematon | tuntematon | tuntematon | tuntematon | 2011-10-21 12:39:49 | |
| switch12 | 192.168.112.1 | tuntematon | tuntematon | tuntematon | tuntematon | tuntematon | 2011-10-21 12:39:52 | |
| switch31 | 192.168.131.1 | up | up | down | down | 5 | 2011-10-21 11:46:55 | |
| switch32 | 192.168.132.1 | tuntematon | tuntematon | tuntematon | tuntematon | tuntematon | 2011-10-21 12:39:53 | |
| switch51 | 192.168.151.1 | up | up | up | up | 5 | 2011-10-21 12:19:12 | |
| switch52 | 192.168.152.1 | up | up | up | up | 5 | 2011-10-21 12:19:13 | |
| switch61 | 192.168.161.1 | up | up | up | up | 5 | 2011-10-21 12:08:43 | |
| switch62 | 192.168.162.1 | tuntematon | tuntematon | tuntematon | tuntematon | tuntematon | 2011-10-21 12:39:57 | |

KUVIO 7. Valvontasovelluksen käyttöliittymä

Kuviossa 7 näkyy valvontasovelluksen käyttöliittymä testauksen aikana. Kuvan oton aikana clan11-, clan12-, clan32- ja clan62-reitittimien sisäverkon liittymät sammutettiin laitteiden hallinnasta. Tämän seurauksena kyseisten laitteiden Lan_stat-muutujan arvoksi vaihtui "administratively down" ja laitteen tilaa kuvaavan värillisen nelikulmion väri muuttui punaiseksi. Sammutettujen liityntöjen takana olivat kytkimet switch11, switch12, switch32 ja switch62. Koska kytkimeen liitetty reitittimen portti oli sammutettu, kytkimeen ei saatu yhteyttä. Kun yhteyttä ei saatu, kaikki laitteen tilaa kuvaavat muuttujat saivat arvoksi "tuntematon" ja laitteen tilaa kuvaavan värillisen nelikulmion väri muuttui harmaaksi. Switch32:n sisäverkon liittymään ei ollut kytketty tietokonetta, joten Lan_stat- ja Lan_prot-muuttujien arvo on "down" ja nelikulmion väri keltainen. Muiden laitteiden tila oli normaali. Klikkaamalla laitteen IP-osoitetta pystyttiin ottamaan telnet-yhteys Nutty-ohjelmalla valvottavan laitteen hallintaa varten.




KUVIO 8. Valvontasovelluksen käyttöliittymä matkapuhelimen näytöllä

Valvontasovellusta voidaan käyttää tietokoneen lisäksi älypuhelimella. Kuvio 8 on matkapuhelimen näytöltä otettu kuvankaappaus. Matkapuhelimena oli Nokia 5230, jonka näytön korkeus on vaakatasossa 360 pikseliä ja leveys 640 pikseliä. Selaimena käytettiin Opera Miniä. Kuvasta selviää, että taulukon luettavuus on hyvä, ja kaikki tarpeellinen informaatio mahtuu näytölle. IP-osoitteen klikkaaminen, eli telnet-hallintayhteyden muodostaminen, ei 5230:lla onnistunut. Markkinoilla on kuitenkin useita matkapuhelinkäyttöjärjestelmiä, joten hallintayhteyden muodostaminen saattaa onnistua muulla matkapuhelinalustalla.

Etuna matkapuhelinyhteensopivuudessa on se, että valvontajärjestelmästä voidaan tehdä kaksi versiota. Yksi, kaikki laitteet näyttävä järjestelmä, voi toimia palveluntarjoajan käytössä, ja toinen suppeampi asiakkaan käytössä. Asiakas voi halutessaan tarkistaa oman verkkonsa laitteiden tilan vaikka toiselta puolelta maapalloa.

## 7.2 Valvontajärjestelmän kehittäminen

Vaikka työssä esitelty valvontajärjestelmä on helpohkosti otettavissa hyötykäyttöön, kehitettävääkin on. Ensimmäisenä on saatava järjestelmän tietoturva kuntoon. On huolehdittava, että järjestelmään ja järjestelmän tiedostoihin on pääsy ainoastaan valtuutetuilla henkilöillä. Käyttäjille on luotava turvalliset kirjautumistunnukset, ja valvontajärjestelmän tiedostojen käyttöoikeudet on rajattava siten, että valtuutetuilla käyttäjillä tai käyttäjäryhmillä on pääsy tiedostoihin.

Tällä hetkellä valvontajärjestelmä tukee ainoastaan Ciscon laitteita. Tulevaisuudessa olisi hyvä lisätä tuki muidenkin valmistajien, kuten HP:n tai Huawein, laitteille. Käytännössä uusien laitemerkkien tuen lisääminen on kohtalaisen helppoa. Muokkaamalla hyvin kommentoituja tiedostoja voidaan tuki uudelle laitemerkille lisätä hyvin nopeasti.

Uuden laitteen lisääminen valvontasovellukseen vie ajassa mitattuna ainoastaan muutaman hetken, mutta toimenpiteenä lisääminen on hieman hankalaa. Ensin on kirjauduttava tietokantaan ja lisättävä laitteen tunnus ja IP-osoite käsin. Tämän jälkeen on luotava laitteelle oma kansio ja siirrettävä kansioon pari tiedostoa ja lopuksi muokattava paria tiedostojen riviä. Laitteiden lisäämistä varten olisi luotava erillinen PHP:lla luotu nettisivu, jonka kautta lisääminen olisi helppoa. Nettisivulla voisi olla alasvetovalikot, joista valitaan laitteen merkki ja verkkoliittymien tunnukset. Laitetunnuksen ja IP-osoitteen lisäämistä varten olisi tekstilaatikko. Syötettyjen tietojen perusteella PHP-skripti pystyisi lisäämään laitteen MySQL-tietokantaan, luomaan laitteelle kansion ja muuttamaan kansion käyttöoikeudet. Kansioon siirrettävien tiedostojen muokkaaminen ja käyttöoikeuksien vaihtaminenkin onnistuu PHP:llä.

Valvontajärjestelmä ei automaattisesti hae ja tallenna laitteiden tietoja, vaan automaattinen haku ja tallennus on toteutettavissa cron-ajastuspalvelulla. Croniin lisätään viittaus erilliseen tiedostoon, jossa ovat varsinaiset ajettavat komennot. Näin laitteiden lisääminen ja poistaminen listasta on helppoa. Listaa voidaan muokata automaattisesti PHP:llä esimerkiksi edellisessä kappaleessa esitetyllä

nettisivulla. Jos käyttäjälle näkyvää valvontasivua halutaan päivittää automaattisesti, päivittäminen voidaan suorittaa lisäämällä pari JavaScript-koodiriviä.

Hälytykset, kuten linkkien tippuminen, ilmoitetaan käyttäjälle taulukon rivillä näkyvällä värillisellä nelikulmiolla. Samalta riviltä näkee yhdellä silmäyksellä, mikä tilanne on aiheuttanut hälytyksen. Hälytysten havaitsemista voidaan kehittää eteenpäin esimerkiksi lähettämällä sähköposti tai luomalla erillinen taulukko hälytystilassa olevista laitteista. Sähköpostin lähettäminen ja uuden taulukon lisääminen onnistuu PHP:llä, joten erillisiä ohjelmointikieliä ei tarvitse käyttää. Korjaustapahtumista voidaan luoda toinen tietokantataulu. Tauluun tallennetaan korjaustoimenpiteet, jolloin laitteiden kuntoa voidaan valvoa pidemmällä aikavälillä, ja lyhentää jatkossa korjauksiin kuluvaa aikaa. Yhteen tietokantatauluun voidaan koota tiedot laitteesta ja laitteen sijoituspaikasta. Tätä taulua  voidaan käyttää esimerkiksi laskutukseen ja konfiguraatiotietojen tallennukseen.

# 8 YHTEENVETO

Opinnäytetyön aiheena oli luoda aloittelevalle PK-yritykselle verkonvalvontajärjestelmä, jolla pystyy valvomaan joko yrityksen omia laitteita tai verkon rajalla olevia asiakkaiden laitteita. Tavoitteena oli luoda järjestelmä siten, että valvontajärjestelmän alustan ohjelmistoiksi valittaisiin maksuttomia avoimeen lähdekoodiin perustuvia sovelluksia, ja itse valvontasovellus ohjelmoitaisiin itse. Toisena tavoitteena oli luoda järjestelmä sellaiseksi, että ylläpito onnistuisi helpohkosti, ja järjestelmän muuttaminen yrityksen tarpeita vastaavaksi olisi mahdollista.

Verkonvalvonnassa voidaan puhua geneerisestä mallista, kolmesta eri verkonvalvonnan yhteyskäytännöstä, eli komentoliittymästä, SNMP:stä ja Syslogista, sekä FCAPSista, joka on ISO Telecommunications Management Network -protokollamalli ja kehys tietoliikenneverkkojen hallintaan. FCAPSin avulla voidaan määritellä, mitä ja miten verkkoa voidaan valvoa ja hallita, jotta verkko toimisi toivotulla tavalla.

Käytännön osuus aloitettiin kuvaamalla valvontajärjestelmän toiminta. Työssä on kuvattu valvontajärjestelmän pystyttäminen mahdollisimman tarkasti aina käyttöjärjestelmän asentamisesta lähtien. Valvontajärjestelmä hakee valvottavasta laitteesta tiedot Expect-skriptin avulla telnet- tai SSH-yhteyden yli ja tallentaa halutut tiedostot tekstitiedostoon. PHP-skripti jäsentää tekstitiedostosta haetut tiedot laitekohtaisesti MySQL-tietokantaan.

Jokaisesta laitteesta haetaan sisä- ja ulkoliittymän linjaprotokollan tilan ja fyysisen tilan lisäksi prosessorin käyttöaste viimeisen viiden minuutin aikana. Edellä mainittuihin muuttujiin päädyttiin siksi, että kyseiset muuttujat sisältävät laitteiden kannalta oleellisimmat tiedot, ja tietojen pitäisi olla saatavilla lähes jokaisesta laitteesta. Käyttäjä pääsee valvomaan laitteiden tilaa verkkoselaimella. Verkkosivu on toteutettu PHP:llä, jonka avulla haetaan MySQL-tietokannasta laitteiden tiedot ja luodaan tiedoista helposti luettava taulukko. Taulukossa jokaisen rivin lopussa on värillinen nelikulmio, joka kuvaa laitteen tilaa. Esimerkiksi vihreä nelikulmio tarkoittaa, että laite on kunnossa, ja punainen viittaa vakavaan vikaan. Web-pohjainen toteutus valittiin laitteistoriippumattomuuden ja helppokäyttöisyyden

vuoksi. Valvontasovellukseen voidaan täten luoda esimerkiksi asiakkaalle oma käyttöliittymänsä, josta hän voi tarkistaa oman verkkonsa laitteiden tilan matkapuhelimensa verkkoselaimella.

Verkonvalvontajärjestelmää testattiin Lahden ammattikorkeakoulun tekniikan alan tietoverkkolaboratoriossa. Järjestelmään syötettiin toistakymmentä valvottavaa kytkintä ja reititintä. Testausta varten luotiin yksinkertainen skripti, jonka avulla laitteista haettiin halutut tiedot, jotka tallennettiin tietokantaan. Toimivuutta testattiin katkaisemalla laitteiden välisiä linkkejä, ja tutkimalla valvontasovelluksella toimien tuloksia. Tuloksista ei löytynyt yllätyksiä, joten valvontajärjestelmä todettiin toimivaksi.

Verkonvalvontasovellusta voidaan kehittää eteenpäin. Uusien laitteiden lisäämiseen ei mene muutamaa hetkeä kauempaa, mutta työ on hieman monimutkaista. Kannattavaa olisi luoda erillinen verkkosivu PHP:llä, jonka avulla laitteiden lisääminen onnistuisi samaan tapaan kuin yksinkertaisen lomakkeen täyttäminen. Tällä hetkellä valvontasovellus tukee ainoastaan Ciscon laitteita, joten tuki muidenkin valmistajien laitteille olisi hyvä lisätä.

Toteutettu verkonvalvontajärjestelmä on toimivaksi todettu runko pienemmän verkon valvontaan. Järjestelmä voitaisiin periaatteessa ottaa suoraan käyttöön, kunhan tietoturvasta ensin huolehdittaisiin. Kannattavinta on kuitenkin muokata valvontasovellus oman yrityksen tarpeisiin sopivaksi ennen järjestelmän ottamista tuotantokäyttöön. Verkonvalvontajärjestelmästä pystyy kehittämään myös erillisen tuotteen, jossa laitteisto ja ohjelmisto on yhdistetty. Valmista tuotetta voitaisiin kaupata muiden yritysten käyttöön, jotka haluaisivat hankkia verkkoonsa valmiin verkonvalvontajärjestelmän. Tällöin oman yrityksen liikeideaksi tulisi valvontajärjestelmän kehittäminen ja kauppaaminen.

Verkonvalvonnan avulla yritys voi valvoa verkossa sijaitsevia aktiivilaitteita, puuttua nopeasti häiriötilanteisiin ja kerätä historiatietoa esimerkiksi ennustamaan tulevaisuuden laitehankintoja. Verkonvalvonnan avulla voidaan säästää myös sähköä. Tarkkailemalla laitteiden, kuten IP-puhelinten tai kytkimien, käyttöasteita voidaan luoda energian säästöön pyrkiviä käytäntöjä. Käytäntöjen avulla voidaan

sammuttaa laitteet, kun työntekijät esimerkiksi poistuvat työtilasta. Sähköä säästämällä yrityksen sähkölasku pienenee ja autetaan säilyttämään maapallon uusiutumattomia energiavaroja.

LÄHTEET

Akin, T. 2002. Hardening Cisco Routers. Sebastopol: O'Reilly & Associates.

Claise, B. & Wolter, R. 2007. Network Management: Accounting and Performance Strategies. Indianapolis: Cisco Press.

Clemm, A. 2007. Network Management Fundamentals. Indianapolis: Cisco Press.

Converse, T., Park, J. & Morga, C. 2004. PHP5 and MySQL Bible. Indianapolis: Wiley.

Deveriya, A. 2006. Network Administrators Survival Guide. Indianapolis: Cisco Press.

Ding, J. 2009. Advances in Network Management. Boka Raton: CRC Press.

Dyer, R. 2008. MySQL in a Nutshell. Second Edition. California: O'Reilly Media.

Goralski, W. 2009. The Illustrated Network: How TCP/IP Works in a Modern Network. Burlington: Morgan Kaufmann Publishers.

Hill, B., Helmke, M., Graner, A. & Burger C. 2011. The Official Ubuntu Book. Sixth Edition. New Jersey: Prentice Hall.

Kamber, H. 2006. Data Mining: Concepts and Techniques. Second Edition. Burlington: Morgan Kaufmann Publishers

Kofler, M. 2005. The Definite Guide to MySQL 5. Third Edition. New York City: Apress.

Libes, D. 1994. Exploring Expect. California: O'Reilly Media.

Musciano, C & Kennedy, B. 2006. HTML & XHTML: The Definite Guide. Sixth Edition. California: O'Reilly Media.

Neumann, J. C. 2009. Cisco Routers for the Small Business: A Practical Guide for IT Professionals. New York City: Apress.

Olsson, T. & O'Brien, P. 2008. The Ultimate CSS Reference. Melbourne: Site-point.

Plevyak, J & Sahin, V. 2010. Next Generation Telecommunications Networks, Services and Management. New Jersey: Wiley-IEEE Press.

Verma, D. 2009. Principles of Computer Systems and Network Management. New York: Springer.

LIITTEET

Liite 1. Tietojen_haku.exc-esimerkkitiedosto

```
#!/usr/bin/expect

set timeout 30

set LaiteID "Asiakas_1"
set host "192.168.11.1"
set password "cisco"
set enablePassword "cisco"
set wanID "ATM0"
set lanID "FastEthernet0"

log_file -noappend $LaiteID.txt

spawn telnet $host

expect "Password:" {
 send "$password\n"
 }

expect ">" {
 send "enable\n"
 }

expect "Password:" {
 send "$enablePassword\n"
 }

expect "#" {
 send "show ip interface $wanID | include $wanID\n"
 }

expect "#" {
 send "show ip interface $lanID | include $lanID\n"
 }

expect "#" {
 send "show processes cpu | include CPU utilization\n"
 }

expect "#" {
 send "exit \n\r"
 }

 expect eof
```

Liite 2. Valvonta.php-tiedosto

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

 <head>
  <link rel="stylesheet" type="text/css" href="tyylit.css" />
  <title>Valvonta</title>
 </head>

 <body>
  <?php include("mysql-esitys.php") ?>

 </body>

</html>
```

Liite 3. Tyylit.css-tiedosto

```
body {
            background-color: #fbfdff;
}

caption {
            text-align: left;
            color: #323232;
            font-size: larger;
            margin-bottom: 10px;
}

table {
            border-collapse: collapse;
}


tr.otsikot {
            background-color: #3A4856;
}

tr.otsikot:hover {
            background-color: #3A4856;
}

tr.parillinen {
            background-color: #F3F5F7;
}

tr.pariton {
            background-color: #F0F2F4;
}

th {
            padding: 5px;
            color: #FFFFFF;
            font-weight: 100;
}

td {
            padding: 5px 10px;
}

table, th, td {
            border: 1px solid #3A4856;
}


tr:hover {
            background-color: #EAECEE;
}

td.ok {
            background-color: #80c080;
}
```

Liite 3. (jatkuu)

```
td.varoitus {
            background-color: #ffff80;
}

td.halytys {
            background-color: #ff8080;
}

td.tuntematon {
            background-color: #c0c0c0;
}

a {
            color: #3A4856;
            text-decoration: none;
            border-bottom: 1px solid #C6C8CB;
}

a:visited {
            color: #3A4856;
}

a:hover {
            color: #3A4856;
}
```

Liite 4. Vakiomuuttujat.php-esimerkkitiedosto

```php
<?php

// mysql-palvelimen muuttujat
$mysql_palvelin = 'localhost';      // mysql-palvelimen osoite
$mysql_kayttun = 'valvomo';         // mysql-kayttajatunnus
$mysql_salasana = 'passu';          // mysql-salasana
$mysql_tietokanta = 'Valvonta';     // kaytettava tietokanta
$mysql_taulu = 'Laitteet';          // kaytettava tietokantataulu

// laitetietojen muuttujat
$LaiteID = "Operaattori_1";         // laitteen uniikki tunnus
$wanID = "ATM0";                    // ulkoverkon liittyman tunnus
$lanID = "FastEthernet0";           // sisaverkon liittyman tunnus

// virhemuuttujat
// poikkeavasta laitteen tilasta voidaan maarata
// tietty virhepistemaara
// kun pistemaara ylittyy, nostetaan vakavuustasoa

$varoitustaso = 1;      // kun virhepistemaara nousee talle
                        // tasolle, annetaan varoitus

$halytystaso =  3;      // kun virhepistemaara nousee talle
                        // tasolle, annetaan halytys

$Wan_statDown = 3;      // kun ulkoverkon liittyman tila on
                        // down, annetaan nama pisteet

$Wan_statAdmDown = 3;   // kun ulkoverkon liittyman tila on
                        // administratively down, annetaan nama
                        // pisteet

$Wan_protDown = 3;      // kun ulkoverkon liittyman protokollan
                        // tila on down, annetaan nama pisteet

$Lan_statDown = 1;      // kun sisaverkon liittyman tila on
                        // down, annetaan nama pisteet

$Lan_statAdmDown = 3;   // kun sisaverkon liittyman tila on
                        // administratively down, annetaan nama
                        // pisteet

$Lan_protDown = 1;      // kun sisaverkon protokollan tila on
                        // down, annetaan nama pisteet

$CPUyliRajan = 3;       // kun prosessorin kayttoaste viimeisen
                        // viiden minuutin aikana yli $CPUraja :n,
                        // annetaan nama pisteet

$CPUraja = 60;          // cpu:n kayttoasteen prosenttimaara
                        // viim. 5 min aikana, milloin
                        // halytys annetaan

?>
```

Liite 5. Mysql-esitys.php-tiedosto

```php
<?php

include "vakiomuuttujat.php";    // tuodaan vakiomuuttujat

// otetaan yhteys tietokantaan
$yhteys = mysql_connect($mysql_palvelin , $mysql_kayttun ,
$mysql_salasana);
if (!$yhteys) die ('Ei voitu yhdistaa: ' . mysql_error());

// valitaan  tietokanta
$valittu_tietokanta = mysql_select_db($mysql_tietokanta, $yhteys);
if (!$valittu_tietokanta) die ('Ei voitu yhdistaa tietokantaan: '
. mysql_error());

// lahetetaan pyynto, jolla haetaan koko taulun sisalto
$sql = "SELECT * FROM $mysql_taulu ORDER BY LaiteID";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen hakeminen taulusta epaonnistui: ' .
mysql_error());

// taulun sarakkeiden maara
$sarakemaara = mysql_num_fields($tulos);

echo '<table border="1">' . '<caption>' . $mysql_tietokanta .
": " . $mysql_taulu . '<tr class="otsikot">';

// tulostetaan taulukon sarakeotsikot
for($i=0; $i<$sarakemaara; $i++)
{
    $sarakeOts = mysql_fetch_field($tulos);
    echo "<th>{$sarakeOts->name}</th>";
}
echo "<th>Tilanne</th>";         // lisataan yksi sarake
echo "</tr>\n";

// tulostetaan tietokannan rivit, kerataan virhepisteet ja
//tulostetaan virhetulos
$j = 0;
while($rivi = mysql_fetch_row($tulos))
{
   $virhemaara = 0;

   $i = 0;
   while ($i < count($rivi))
   {
     // verrataan tunnettua saraketta ja rivilla samassa
     // sarakkeessa olevaa arvoa toisiinsa ja lisataan
     // virhepisteita tarvittaessa.
     // Esimerkiksi ensimmaisessa casessa $i==2 tarkoittaa
     // tietokantataulun kolmatta saraketta (Wan_stat) ja
     // $rivi[$i] tarkoittaa ensimmaisella rivilla yhta
     // muuttujan arvoa.
     // Ensimmaisessa casessa virhemaaraa lisataan, jos
     // Wan_stat-sarakkeen kohdalla on prosessoitavalla
     // rivilla arvo down
```

Liite 5. (jatkuu)

```
    switch ($rivi[$i])
    {
          case ($i==2 && $rivi[$i]=="down"):
           $virhemaara = ($virhemaara + $Wan_statDown);
           break;

          case ($i==2 && $rivi[$i] == "administratively down"):
           $virhemaara = ($virhemaara + $Wan_statAdmDown);
           break;

          case ($i==3 && $rivi[$i]=="down"):
           $virhemaara = ($virhemaara + $Wan_protDown);
           break;

          case ($i==4 && $rivi[$i]=="down"):
           $virhemaara = ($virhemaara + $Lan_statDown);
           break;

          case ($i==4 && $rivi[$i]=="administratively down"):
           $virhemaara = ($virhemaara + $Lan_statAdmDown);
           break;

          case ($i==5 && $rivi[$i]=="down"):
           $virhemaara = ($virhemaara + $Lan_protDown);
           break;

          case ($i==6 && $rivi[$i]>=$CPUraja):
           $virhemaara = ($virhemaara + $CPUyliRajan);
           break;

          // jos yhteytta valvottavaan laitteeseen ei ole saatu,
          // tietokannassa kaikissa kohdissa laitetta on arvo
          // "tuntematon"
          case ($i==2 && $rivi[$i]=="tuntematon"):
           $virhemaara = -1;
           break;

          default:
    }

  $i++;
  }

  // tuotetaan vakavuustaso
  switch (true)
  {
          case ($virhemaara < $varoitustaso && $virhemaara >= 0):
            $vakavuustaso = "ok";
            break;

          case ($virhemaara >= $varoitustaso && $virhemaara <
          $halytystaso):
            $vakavuustaso = "varoitus";
            break;

          case ($virhemaara >= $halytystaso):
            $vakavuustaso = "halytys";
            break;
```

Liite 5. (jatkuu)

```
            case ($virhemaara == -1):
              $vakavuustaso = "tuntematon";

            default:
              $vakavuustaso = "tuntematon";
    }

    // joka toiselle riville annetaan class="pariton", joka
    // toiselle "parillinen", taulukon ulkoasun muokkaamista varten
    if ($j % 2)
        echo '<tr class ="pariton">';
    else
        echo '<tr class="parillinen">';

    // $rivi on taulukko, josta muuttujat eritellään
    // $solu-muuttujiin ja tulostetaan taulukkoon
    foreach($rivi as $solu)
    {
       // jos muuttujassa on 3 pistetta (ip-osoite), luodaan linkki
       if (substr_count($solu,".")==3)
            echo '<td><a href="telnet://' . $solu . '">' . $solu .
            '</a></td>';
       else
            echo "<td>$solu</td>";
    }

    // solun class-arvo maaritellaan $vakavuustaso :n mukaan
    echo '<td class="' . $vakavuustaso . '"> </td>';
    echo "</tr>\n";
    $j++;
}

echo "</table>";

mysql_free_result($tulos);
mysql_close($yhteys);

?>
```

Liite 6. MySQL-tallennus.php-tiedosto

```php
<?php

include "vakiomuuttujat.php";    // tuodaan vakiomuuttujat

// laitteen tila -muuttujat
// seuraaviin muuttujiin haetaan myohemmin ohjelmassa tiedot,
// nyt vain luodaan muuttujat
$Wan_stat = "";       // ulkoverkon liittyman tila
$Wan_prot = "";       // ulkoverkon liittyman protokollan tila
$Lan_stat = "";       // sisaverkon liittyman tila
$Lan_prot = "";       // sisaverkon protokollan tila
$CPU = "";            // prosessorin kayttoaste viimeisen
                      // viiden minuutin aikana

// lisataan .txt-paate laitteen ouput-tiedostonimen
// rakentamista varten
$LaiteIDtxt = $LaiteID . ".txt";

// *****
// tiedotMuuttujiin-funktio tallentaa ylla luotuihin muuttujiin
// laitteen output-tiedostosta haetut tiedot
// *****
function tiedotMuuttujiin()
{
   // tuodaan "kayttajan maariteltavat muuttujat"
   // NAITA MUUTTUJIA EI SAA MUOKATA, AINOASTAAN LUKUKAYTTOON
   global $LaiteIDtxt;
   global $wanID;
   global $lanID;

   //tuodaan "laitteen tila -muuttujat"
   // naihin muuttujiin haetaan halutut tiedot
   global $Wan_stat;
   global $Wan_prot;
   global $Lan_stat;
   global $Lan_prot;
   global $CPU;

   // avataan expect-skriptin luoma tekstitiedosto
   $tiedosto = fopen($LaiteIDtxt,"r") or exit
   ("Tiedostoa $LaiteIDtxt ei voitu avata.");

   while(!feof($tiedosto))
   {
       // haetaan ouput-tiedostosta yksi rivi, ja tallennetaan
       // se sana kerrallaan taulukkoon
       $rivi = (fgetcsv($tiedosto,0," "));
       $elemMaara = count ($rivi);    // elementtien maara
                                      // taulukossa

       $i = 0;
       $j = 1;
```

Liite 6. (jatkuu)

```
// haetaan "laitteen tila -muuttujiin" arvot tekstitiedostosta
// vertaamalla taulukosta kahta perakkain olevaa
// muuttujaa ennalta maarattyyn merkkijonoon
if ($elemMaara > 1)
{
   while ($j < $elemMaara)
   {
      switch ($rivi[$i] . $rivi[$j])
      {
         case ($wanID . "is"):
           // tallennetaan $Wan_stat -tila
           //
           // esim. "ATM0""is""up" eli haluttu tieto
           // on seuraava merkkijono
           $k = $j+1;
           $Wan_stat = $rivi[$k];
           // poistetaan pilkku "up," ja "down," -sanojen
           // lopusta
           $Wan_stat = str_replace("," , "" , $Wan_stat);
           // korjataan tila kokonaiseksi
           if ($Wan_stat=="administratively") $Wan_stat=
           "administratively down";
           // tallennetaan $wan_prot -tila
           $temp = array_search("protocol",$rivi);
           $temp = $temp+2;  // esim. "protocol""is""up"
           $Wan_prot = $rivi[$temp];
           break;

         case ($lanID . "is"):
           // tallennetaan $Lan_stat -tila
           //
           // esim. "FastEthernet0""is""up" eli haluttu tieto
           // on seuraava merkkijono
           $k = $j+1;
           $Lan_stat = $rivi[$k];
           // poistetaan pilkku "up," ja "down," -sanojen
           // lopusta
           $Lan_stat = str_replace("," , "" , $Lan_stat);
           // korjataan tila kokonaiseksi
           if ($Lan_stat=="administratively") $Lan_stat=
           "administratively down";
           // tallennetaan $Lan_prot -tila
           $temp = array_search("protocol",$rivi);
           $temp = $temp+2;  // esim. "protocol""is""up"
           $Lan_prot = $rivi[$temp];
           break;

          case ("five" . "minutes:"):
            // tallennetaan $CPU -tila
            //
            // esim. "five""minutes:""3%" eli haluttu tieto on
            //seuraava merkkijono
            $k = $j+1;
            $CPU = $rivi[$k];
            // poistetaan prosenttimerkki luvun lopusta
```

Liite 6. (jatkuu)

```
                $CPU = str_replace("%" , "" , $CPU);
                break;

              // jos yhteytta ei saada, annetaan arvo "tuntematon"
              // virheviestit ovat jokaisella
              // kayttojarjestelmalla ja protokollalla erilaisia
              // alla oleva tulee linuxilla telnet-yhteyden
              // epaonnistuessa
              case ("Unable" . "to");
                $Wan_stat = "tuntematon";
                $Wan_prot = "tuntematon";
                $Lan_stat = "tuntematon";
                $Lan_prot = "tuntematon";
                $CPU = "tuntematon";
                break;

              default:
            }
            $i++;
            $j++;
          }
        }
    }
fclose($tiedosto);
}

// *****
// tiedotTietokantaan-funktio tallentaa
// "laitteen tila -muuttujat" mysql-tietokantaan
// *****
function tiedotTietokantaan()
{
   // tuodaan kayttajan maariteltavat muuttujat
   global $LaiteID;

   // tuodaan mysql-muuttujat
   global $mysql_palvelin;
   global $mysql_kayttun;
   global $mysql_salasana;
   global $mysql_tietokanta;
   global $mysql_taulu;

   // tuodaan laitteen tila -muuttujat
   global $Wan_stat;
   global $Wan_prot;
   global $Lan_stat;
   global $Lan_prot;
   global $CPU;

   // otetaan yhteys mysql-palvelimeen
   $yhteys = mysql_connect($mysql_palvelin , $mysql_kayttun ,
   $mysql_salasana);
   if (!$yhteys) die ('Ei voitu yhdistaa: ' . mysql_error());
```

Liite 6. (jatkuu)

```
// valitaan tietokanta
$valittu_tietokanta = mysql_select_db($mysql_tietokanta,
$yhteys);
if (!$valittu_tietokanta) die ('Ei voitu yhdistaa
tietokantaan: ' . mysql_error());


// haetaan mysql-taulusta sarakkeiden otsikot
// lahetetaan pyynto, jolla haetaan koko taulun sisalto
$sql = "SELECT * FROM $mysql_taulu";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen hakeminen taulusta
epaonnistui: ' . mysql_error());
// taulun sarakkeiden maara
$sarakemaara = mysql_num_fields($tulos);
$j=1;
for($i=0; $i<$sarakemaara; $i++)
{
    // hakee sarakkeen otsikon ominaisuudet
    $sarakeOts = mysql_fetch_field($tulos);
    // hakee taulukkoon sarakeotsikon otsikon
    $otsikkoTaulukko[$j] = $sarakeOts->name;
    $j++;
}

// viedaan laitteen tila -muuttujista tiedot tauluun
// $otsikkoTaulukko[1] tarkoittaa mysql-taulun
// ensimmaisen sarakkeen otsikkoa jne.
// esimerkkina ensimmainen komento menee selvakielisena nain:
// UPDATE Laiteet SET Wan_stat='up' WHERE
// LaiteID='Operaattori_1'
$sql = "UPDATE $mysql_taulu SET $otsikkoTaulukko[3] =
'$Wan_stat' WHERE $otsikkoTaulukko[1] = '$LaiteID'";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen syottaminen epaonnistui: ' .
mysql_error());

$sql = "UPDATE $mysql_taulu SET $otsikkoTaulukko[4] =
'$Wan_prot' WHERE $otsikkoTaulukko[1] = '$LaiteID'";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen syottaminen epaonnistui: ' .
mysql_error());

$sql = "UPDATE $mysql_taulu SET $otsikkoTaulukko[5] =
'$Lan_stat' WHERE $otsikkoTaulukko[1] = '$LaiteID'";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen syottaminen epaonnistui: ' .
mysql_error());

$sql = "UPDATE $mysql_taulu SET $otsikkoTaulukko[6] =
'$Lan_prot' WHERE $otsikkoTaulukko[1] = '$LaiteID'";
$tulos = mysql_query($sql, $yhteys);
if (!$tulos) die('Tietojen syottaminen epaonnistui: ' .
mysql_error());
```

Liite 6. (jatkuu)

```
    $sql = "UPDATE $mysql_taulu SET $otsikkoTaulukko[7] = '$CPU'
    WHERE $otsikkoTaulukko[1] = '$LaiteID'";
    $tulos = mysql_query($sql, $yhteys);
    if (!$tulos) die('Tietojen syottaminen epaonnistui: ' .
    mysql_error());

    mysql_close($yhteys);
}


// kutsutaan tiedotMuuttujiin-funktiota, jotta saadaan
// tallennettua "laitteen tila -muuttujiin" tiedot
tiedotMuuttujiin();

// kutsutaan tiedotTietokantaan-funktiota, jotta saadaan
// tallennettua "laitteen tila -muutujat" mysql-tietokantaan
tiedotTietokantaan();

?>
```